A Study on Confidentiality and Authenticity of Document using Public Key Cryptography

January 2008

Yoshiki Sameshima

A Study on Confidentiality and Authenticity of Document using Public Key Cryptography

Submitted to

Graduate School of Information Science and Technology

Osaka University

January 2008

Yoshiki Sameshima

Author's Publications for Doctoral Degree Application

A. Journal Papers

1. Yoshiki Sameshima and Peter Kirstein, Secure Document Interchange: A Secure User Agent, *Computer Networks and ISDN Systems*, vol.28, no.4, pp.513-523, 1996.
2. Yoshiki Sameshima and Peter Kirstein, Authorization with Security Attributes and Privilege Delegation: Access Control beyond the ACL, *Computer Communications*, vol.20, no.5, pp.376-384, 1997.
3. Yoshiki Sameshima and Tsutsumi Tsutsumi, Reducing Certificate Revocation and Non-Repudiation Service in Public Key Infrastructure, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol.E83-A, no.7, pp.1441-1449, 2000.
4. Yoshiki Sameshima, Hideaki Saisho, Kazuko Oyanagi, and Tsutomu Matsumoto, Multiparty DSA Signature Generation without Simultaneous User Operations, *IEICE Transaction on Information and Systems*, vol. E87-D, no.8, pp.2095-2105, 2004.

B. International Conference Papers

1. Yoshiki Sameshima and Peter Kirstein, Secure Document Interchange - A Secure User Agent, in *Proceedings of TERENA, 6th Joint European Networking Conference*, pp.323-1-10, 1995.
2. Yoshiki Sameshima, Security Architecture based on Secret Key and Privilege Attribute Certificates, in *Proceedings of the IFIP/IEEE International Conference on Distributed Platforms*, pp.357-369, 1996.
3. Yoshiki Sameshima, A Key Escrow System of the RSA Cryptosystem, in *Proceedings of the First International Information Security Workshop 1997 (ISW'97)*, pp.135-146, 1997.
4. Yoshiki Sameshima, Hideaki Saisho, Tsutomu Matsumoto, and Norihisa Komoda, Windows Vault: Prevention of Virus Infection and Secret Leakage with Secure OS and Virtual Machine, in *Pre-Proceedings of the 8th International Workshop of Information Security Applications 2007 (WISA 2007)*, pp.249-261, 2007.

C. Domestic Conference and SIG Papers

1. Y. Sameshima, Problems and Solution of X.509 Authentication Framework (in Japanese), in *Proceedings of the 1997 Symposium on Cryptography and Information Security*, 8B, 1997.
2. Y. Sameshima and H. Miyazaki, Privacy Enhanced Message System using Secret-Key and User-Attribute Certificates (in Japanese), in *Proceedings of the 5th Workshop on Multimedia and Distributed Systems*, pp.85-92, 1997.
3. Y. Sameshima, One Round (k, n) Threshold Nyberg-Rueppel Signature Scheme (in Japanese), in *Proceedings of the 2000 Symposium on Cryptography and Information Security*, C25, 2000.
4. Y. Sameshima, H. Saisho, and T. Matsumoto, Multiparty DSA Signature Generation without Concurrent Processing (in Japanese), in *Technical Report of IEICE*, ISEC 2001-66, pp.97-104, 2001.
5. H. Saisho, Y. Sameshima, and T. Matsumoto, Efficiency Considerations for Multiparty DSA Signature Generation System using Smart Cards (in Japanese), in *Proceeding of Computer Security Symposium 2001*, pp.349-354, 2001.
6. Y. Aoyagi and Y. Sameshima, Secret Information File Sealing System (in Japanese), in *Proceeding of Computer Security Symposium 2002*, pp.59-64, 2002.
7. Y. Sameshima, H. Saisho, and T. Matsumoto, Fast Large Prime Number Generation using Multi-le-Precision Integer Operations (in Japanese), in *Proceedings of the 2002 Symposium on Cryptography and Information Security*, pp.61-66, 2002.
8. Y. Nakamura and Y. Sameshima, Configuration System for Access Control Policy of Security-Enhanced Linux (in Japanese), in *Proceedings of the 2003 Symposium on Cryptography and Information Security*, pp.831-836, 2003.
9. Y. Nakamura and Y. Sameshima, Log Audit Tool for SELinux (in Japanese), in *Proceedings of the 2004 Symposium on Cryptography and Information Security*, pp.293-298, 2004.
10. Y. Sameshima and H. Saisho, Prevention of Informatino Leakage and Virsu Infection with use of Secure OS and Virtual Machine (in Japanese), in *Proceedings of the 13th Workshop on Multimedia and Distributed Systems*, pp.151-155, 2005.

Summary

This dissertation is a study on confidentiality and authenticity of document using public key cryptography researched through 1992 to 2007 by the author who is enrolled at Hitachi Software Engineering Co., Ltd. and Graduate School of Information Science and Technology, Osaka University.

With the spread of the Internet, threats to confidentiality and authenticity of office document have become to be actual; the most typical example is leakage of customer data by lost of storage media or harmful e-mail from persons. Against such threats there are the following countermeasures. Firstly, as for countermeasure for lost of media, wiretapping, and sender spoofing, technologies such as encryption, authentication, and integrity check of document, are used with public key cryptography based on the Public Key Infrastructure (PKI). Secondly, countermeasures to unauthorized access to documents on a server are authentication of accessing entity and access control on the server. Thirdly, a typical countermeasure for virus is anti-virus software based on pattern matching with signature of known viruses. However, there is also a system which is comprised of two virtual workstations running on a single PC hardware to protect data from unknown viruses; one is used for secret file and the other is used for non-secret file that may contain viruses. Even if the second workstation for non-secret is infected with virus, the first workstation is safe because the two workstations run on separated virtual machines.

However, other problems appear from the point of view of usability and operation when the above technologies are used. Firstly, Multiparty Signature Generation (MSG) is a useful technology to protect a signature key, especially the private key of a Certification Authority (CA) which is the root of trust of the PKI. However, the MSG of Digital Signature Algorithm (DSA) requires simultaneous operations of key holders, and generation process does not fit workflow process which is sequential, not simultaneous. Sequential and one-round MSG is required to realize an efficient signing process. Secondly, the PKI itself has a problem; when a public key certificate is revoked with a reason of personnel change or lost of private key, corresponding signatures become invalid, because the public key is invalid with the revocation. Considering use of signed documents, the signatures should be valid after the public key revocation unless the documents are tampered, and an additional scheme is required to the PKI. Thirdly, along with the two problems of key management, there is another problem of encryption and signing of documents; when a structured

document containing multiple chapters, tables and pages is exchanged in a group, multiple signatures and encryption of such parts are required, but an existing cryptographic envelope does not support such function. Considering document creation by multiple authors and access restriction within the group, an encryption and signing technology for structured documents is required. Fourthly, as for access control to documents on a server, the reference monitor model within a single security domain works well, but the authorization model is not well formulated in a multiple domain environment. A model of privilege delegation crossing domain boundary is required to realize authorization of the Internet scale. Finally the system comprised of two virtual workstations for virus protection is safe against unknown virus, but the user should change workstations according to processing information category and this is burden for the user. A system which does not make end user be aware of information category is desirable.

This dissertation is composed of seven chapters and proposes systems or schemes which satisfy the above requirements. Chapter 1 lists security threats to confidentiality and authenticity of documents, discusses problems of existing countermeasures against the threats, and then describes strategies to solve the problems. Chapter 2 proposes the scheme of MSG of DSA without simultaneous key holders' operations, and performance evaluation of a prototype on a smartcard, security against adaptive chosen message attack, and application to other signature scheme are discussed. Chapter 3 introduces an attribute with validity period and a certificate verification service with time stamp in order to solve the signature invalidation problem after public key certificate revocation. Performance of the service is evaluated as well as security. Chapter 4 discusses security requirements of document interchange, and security of Office Document Architecture (ODA) of ISO standard is introduced. Compatibility problems of the two standards, ODA and PKI, and resolutions are discussed in details, followed by discussion on problems on integration with an existing ODA editor. Chapter 5 treats problems of authorization in multiple domains; after requirements of authorization on a document server are specified, an authorization scheme with the combination of a Privilege Attribute Certificate (PAC) and a Control Attribute Package (CAP), and privilege delegation scheme crossing boundary of domains are proposed and evaluated. Chapter 6 proposes "Windows Vault" which solves the usability problem of the two virtual workstations system. After describing gateways connecting the two workstations realizing safe integration of e-mail clients on the two workstations, performance and security extension of the gateway are discussed. Finally Chapter 7 concludes this study and discusses directions for the future research.

Contents

Chapter 1

Introduction

1.1 Background

In the mid 1990's, when a commercial internet service started, the main security problem was attack from the Internet, such as destruction of web page, denial of service, or virus contained in e-mail message. However, the situation has changed since the personal information protection law [ACT2003]; all organizations including commercial companies and government organizations are required to keep personal information secret and correct. Another aspect of data security is to keep document unchanged for years; organization can store electronic documents instead of printed papers which retention time is regulated, with the guarantee that documents have not been changed since creation [ACT2004].

In general, information security is defined as maintenance of confidentiality, integrity, and availability of information asset. The three security attributes of information are defined as follows:

- Confidentiality
  The information is accessed only by authorized subject which is permitted to access.
- Integrity
  The information and process of the information are accurate and complete.
- Availability
  Authorized subject can access or process the information whenever the information is needed.

The theme of this dissertation is the security of office documents or files which are the most familiar data for end users. In case of office documents, the following attribute is more meaningful than integrity:

- Authenticity
  The information is created by the subject as claimed and has not been changed since it was created.

Afterwards, this research focuses on authenticity instead of integrity except that strict distinction is required.

Table 1.1 shows security threats to office documents, which are classified with the view of three aforementioned attributes and document location. Office documents on a client PC are created not only by the end user; they are sent as attached files of e-mail or retrieved from a document server. Such files are stored in a Universal Serial Bus (USB) memory as well as local hard disk. Office documents on network are not only on line but also on mail servers on which the documents are temporary stored. Office documents on servers are those on a file server on Local Area Network (LAN) or Web server connected to the Internet. In the following, each threat shown in Table 1.1 is described.

Table 1.1: Security Threats to Documents

	Client	Network	Server
Confidentiality	● theft/lost of media ● leakage by virus ● intentional leakage by authorized user	● wiretapping	● leakage by unauthorized access ● leakage by manager
Authenticity	● tamper by virus ● tamper by user	● sender spoofing ● denial of sending ● tamper of communication data	● tamper by unauthorized access ● tamper by manager
Availability	● data lost by failure ● deletion by user ● deletion by virus	● failure of network device/line ● denial of service	● deletion by unauthorized access ● deletion by manager ● denial of service

Theft or lost of storage media is the most typical threat to confidentiality; the memory size of the current storage media is so large that a user can store millions of customer data, and the lost of such information leads to disrepute of the organization.

Some virus leaks local files of PC; the most famous virus leaks the files to a peer-to-peer network, and copies of the files are scattered over the network. It is impossible to delete all the copies and the files can be accessed by everyone forever. There is an intentional leakage by an authorized user; the methods of leakage are use of network, storage media, etc.

A cause of lost of authenticity, strictly integrity, of documents stored on a client PC is virus. Some virus rewrites files and adds itself to the files. An authorized user can rewrite files and change the time of last modification time. Document files are unavailable with some causes; typical reason is failure of hardware or software, but a user may delete files by carelessness. Another cause is a virus; the virus deletes files or encrypts them to kidnap.

Wiretapping is a cause of lost of confidentiality of documents on network; unencrypted network traffic of wireless LAN can be monitored by anyone who can receive the electric wave. People who can access router or switch can monitor the traffic through the monitor ports of the network devices.

Lost of authenticity on network does not only caused by rewrite of communication data; sender spoofing happens more frequently. Typically an office document is sent via e-mail, and the recipient considers that the document is sent from the sender in the 'From' field of the e-mail message. However, there is no authentication mechanism in the mail protocol, SMTP [PJ1982], and the field can be spoofed easily. There is another threat to authenticity. Denial of sending is the threat that the sender denies the fact of sending; even if the recipient shows the message from the sender, the sender insists that the 'From' field is spoofed and she/he did not send the message.

Availability of network is lost typically with failure of network devices or communication lines. Another reason is denial of service attack; the attacker sends lots of bogus network packets to server up to exceed the capacity of the server, and it cannot process regular requests.

If an attacker gets privilege of a network server, confidentiality, authenticity, and availability of the data on the server are lost. This is also true when a malicious operator manages the server.

The main theme of this dissertation is security against the threats to confidentiality and authenticity, because the most serious threat of availability is data lost and this threat is basically covered by data backup. Countermeasures for the two threats are divided into two categories as shown in Table 1.2.

Table 1.2: Protection Methods

Method	Threat	Technology	Problem
Protection by document itself ● Encryption ● Digital signature	● theft/lost of media ● leakage by virus ● intentional leakage by authorized user ● wiretapping ● sender spoofing ● denial of sending ● leakage by unauthorized access ● leakage by manager	Private key protection by dividing	All key holders must sign simultaneously.
		Revocation of public key certificate with CRL	Digital signature becomes invalid with certificate revocation.
		Encryption and signing of document part	Partial encryption and signing are not supported.
Protection outside document ● Access control ● Virus protection	● tamper by virus ● tamper by user ● tamper of communication data ● tamper by unauthorized access ● tamper by manager	Access control with security attribute	Access control in multiple domains is insufficient.
		Protection from unknown virus with system isolation	Multiple client use changes user operation.

The first category is protection by document itself, and two methods belong to this category, encryption and digital signature. The two technologies protect document files from the following threats: theft/lost of storage media, leakage by virus, wiretapping, sender spoofing, denial of sending, intentional leakage by authorized user, leakage by unauthorized access, and leakage by manager. While documents stored on media are encrypted with symmetric encryption algorithm, those transferred between users via network are encrypted with both symmetric and public key encryption algorithms [SB1996]; a document itself is encrypted with a randomly generated symmetric key and the key is encrypted with the public key of the recipient user. As far as the private key of the recipient user is protected, the document is safe. On the other hand, digital signature is the combination of a hash function and a public key encryption; the hash function calculates the fingerprint or hash value of the document, and the signature of the document is calculated with the fingerprint and the private key of the signer. The signature is verified with the public key of the signer.

Document protected with encryption and digital signature is safe, if the authentic public key of recipient or signer is available. In order to get the public key, the public key infrastructure (PKI) [CCITT1988, HR1999] is used. Every user of the PKI trusts a third party, Certification Authority (CA), who distributes the authentic public key through a public key certificate. The certificate contains subject identity, her/his public key, validity time of the certificate, the CA identity, etc., and all the information is digitally signed with the private key of the CA. With verification of the certificate, the correct public key of a user can be obtained and used for encryption and verification of digital signature. However, there are problems inside and around the PKI; in the following, problems of protection of private key, invalidation of digital signature, and security of structured office document are described.

(1) Protection of Private Key by Dividing

Protection of a private key is a serious problem, especially the key of a CA. If the key is stolen, the damage is very enormous, because fake user certificates are created freely, and the infrastructure collapses. A typical technology to protect the signature key is Multiparty Signature Generation (MSG) and a CA introduced the MSG technique [CC2002]; in the MSG scheme, the signature key is divided into multiple pieces which are hold by multiple key holders, and the key holders cooperate to make a signature without revealing the divided keys to the others. Since a valid signature cannot be generated even if there is a malicious key holder, this technique realizes high level security.

While an RSA signature can be generated with one-round sequential operations of the key holders, a signature generation of the Digital Signature Algorithm (DSA) [NIST1998] requires simultaneous operations of the key holders, and a work flow system cannot be used to the signature generation process. This is inconvenient and non-effective from the view of business process within an enterprise. This is the first problem.

(2) Revocation of Public Key Certificate with CRL

A public key certificate itself is a static data and it may include old, incorrect information. The certificate may become invalid with some reasons: lost of private key in case that a user forgets the password protecting the key, or old title and department after personnel change, etc. In such a case, the user requests revocation of the certificate and the CA publishes a Certification Revocation List (CRL) to transfer the information of revoked certificates. Once the certificate is revoked, the signature of the

user becomes invalid, because the public key used for the signature verification becomes invalid. This invalidation of the digital signature in accordance with the certificate revocation is the second problem.

(3) Encryption and Signing of Document Part

Encryption and signing are applied to a whole data basically. However, if a document is composed of multiple pages, paragraphs, figures, etc. and the parts are written by different authors, it is desirable that each part is signed by each author. Moreover, some parts may need access control; limited members are permitted to read the parts. As a result, encryption and signing of parts of document are required. This is the third problem.

The second category is protection outside document, that is, access control to document. Access control is realized mainly by OS, however the current mechanism of access control of the OS is not enough for protection from virus with two reasons. The first reason is that the OS cannot distinguish the accessing entity is a correct user or not. Some virus is contained in a document file and it is activated when a user opens the file. In this case, the subject of the activated virus is the user, and the OS allows the access of the virus to the resources that the user can access. The second reason is vulnerability of OS or application programs. Some virus exploits the vulnerability, and gets privilege of the user of application or the administrator, and the OS fails to distinguish the accessing subject again. With these reasons virus protection software which distinguishes virus is required. But there remain problems in both access control itself and virus protection software.

(4) Access Control with Security Attribute

A typical access control mechanism depends on the subject identity or group, and access type; this is sufficient if both the subject and object belong to the same domain which is a collection of users, computers and other resources that are under a single administration. However, if the subject and object belong to different domains, the access control mechanism does not work, and privilege delegation of subject is required. This is the fourth problem.

(5) Protection from Unknown Virus with System Isolation

The current virus protection software distinguishes virus with pattern matching, and as a result, an unknown virus which pattern is not contained in the pattern database

cannot be detected. As a countermeasure against the unknown virus, there is a system which is comprised of two virtual workstations integrated into a single PC hardware with virtual machine technology; a workstation is used for secret and the other for non-secret or public. The system is very secure because the secure workstation is virtually separated at physical level from the unsecure workstation which may be infected with virus. The fifth problem is that the user needs to distinguish the two workstations and change them according to processing information. It is desired not to change operations from the current PC usage.

As described above, the current technologies to protect confidentiality and authenticity of documents have the five problems from the viewpoint of convenience and flexibility as shown in Table 1.2. In this dissertation, the following problems are resolved while keeping the security of the current technology:

- All divided signature key holders must sign simultaneously.
- Digital signature becomes invalid when the public key certificate is revoked.
- It is impossible to encrypt and sign parts of document.
- Access control in multiple domains is not sufficient.
- User is required to change client operations of multiple virtual workstations system from current PC usage.

1.2 Related Works

In this section, researches related to the technologies mentioned in the previous section, private key protection by dividing, revocation of public key certificate, encryption and signing of document, access control with security attribute, and protection from unknown virus with system isolation are described.

(1) Protection of Private Key by Dividing

In order to protect a private key from a malicious key holder, division of the key is a typical solution, and there are many researches of this technology. As far as RSA cryptosystem [RR1978, RSA1993], the generation of divided keys is very difficult [GN1999, MM1999a], but use of the private key, decryption or signature generation, is straightforward, and researchers focused on restriction of decryption by investigating authority; in a key escrow system, much attention was paid to the restriction on use of deposited decryption key, and one of the restriction measure was division of the key

[MS1992, YY1996, SY1997b].

On the MSG based on the Discrete Logarithm Problem (DLP), Schnorr's type signature can be generated efficiently [SC1989, PC1996, MK2001], but the MSG of DSA requires much computation [CM1993, GR1996]. A signature based on the DLP contains a random number, and the efficiency of signature generation depends on arithmetic relation between the random number and the signature key; the Schnorr's signature contains sum of the two numbers and the signature is generated very efficiently, but the DSA signature contains quotient of the signature key divided by the random number and this makes the generation process complicated and inefficient. Moreover, the process requires simultaneous computation of the key holders, and it is impossible to generate the DSA signature with one-round sequential process of the key holders.

(2) Revocation of Public Key Certificate

There are several problems around certificate revocation of the PKI: size of CRL, timely distribution of revocation information, and invalidation of signature. An approach to the first problem is delta-CRL [ISO1995]; the delta-CRL contains the certificate data revoked after the previous issued CRL. Not all the revoked certificates are contained in the delta-CRL. The delta-CRL is smaller than the ordinary CRL, and the CA can issue the delta-CRL more frequently. Therefore, a user can get timely revocation information and this is a partial answer to the second problem. But the user needs to collect all the delta-CRLs and verify a signature, so the verification cost increases. Another approach to the first problem is Certification Revocation Tree (CRT) [KH1999]. The revoked certification information is represented in the form of a tree a leaf of which corresponds to a revoked certificate, a node to the hash value of the lower level nodes, and the root node is digitally signed. In order to get revocation status of a certificate, a user retrieves partial tree, and the size of the tree is smaller than the CRL. As a result, the CRT is a solution to the second problem.

Another approach is an on-line verification service; the IETF standard, Online Certificate Status Protocol (OCSP) [MM1999b] service receives a request containing a certificate identity, and sends back the certificate status with the signature of the CA. This approach solves the second problem, however, there still remains the third problem.

(3) Encryption and Signing of Document Part

Secure/Multipurpose Internet Mail Extensions (S/MIME) [RB1999] give a content type and an extension for encryption and signing of MIME [FN1996] data; MIME bring a

structure into an e-mail message and it supports multiple parts, which types are mixed, alternative, digest, parallel, and the data types of each part are image, audio, video, and application data. However, S/MIME are not for structured document in general. RSA PKCS#7 envelope [RSA1993b] supports multiple signature to any data, but it does not support partial encryption nor signing.

(4) Access Control with Security Attribute

While an Access Control List (ACL) model implemented on Windows OS and Linux is the most typical access control model and easily understandable, multilevel security of TCSEC [DD1983] based on the Bell-LaPadula model is one of the most secure access control model, and it is used in a military system. There are significant works on authorization or access control in a centralized system other than these two models. Boolean Expression Evaluation [MD1989] introduced a generalized policy free access control mechanism, and a unified solution of Mandatory Access Control (MAC) and Discretionary Access Control (DAC) policies is proposed in [MC1990]. However, they cannot be extended to a distribution system environment in a straightforward way, because delegation of privilege is out of scope and the representation of authorization information is simple and does not have enough ability to express the semantics of privileges of different security domains.

The ACL scheme, which makes the authorization based on user's identity or group, fits an environment where the number of users is relatively small, such as a local area network in an office. However, in the environment of an organizational scale network to which thousands of hosts are connected or a much bigger scale network such as the Internet, the ACL scheme may not be appropriate because the authorization may be required to depend on not only the user's identity but also various information such as user's title/role, network location, privilege class, and access time.

The OSF/DCE security architecture [YH1995] and the Secure European System for Application in a Multivendor Environment (SESAME) [KP1994] have adopted the Privilege Attribute Certificate (PAC), which contains user's privileges, restrictions on the privileges and identifiers for auditing and charging, and the PAC is well structured to transmit authorization information of the user. However, the architectures do not specify the access control information of objects being accessed such as files, application entities, nor how the authorization is made.

(5) Protection from Unknown Virus with System Isolation

Current approach to detect unknown virus is to monitor its behavior. However, the

emergence of targeted attack may make this approach may become less effective, because the attacker tunes the virus behavior not to be detected by the anti-virus software of the target user. Another reason is the number of viruses is small and the viruses may not be detected by observation network of a virus vendor.

NetTop [HP2004, MR2000] is a composite of multiple virtual workstations and safe against unknown virus because each workstation is virtually separated with virtual machine and secure OS. But the target of NetTop is mainly intelligence community; the user of NetTop is required to have awareness of data isolation or multilevel security.

1.3 Research Strategies

As shown in Section 1.1, there are two categories of measurements for security threats of document, but each technology realizing the measurements has problem in usability, flexibility or coverage. Solutions against the problems are proposed, which relations to the problems are shown in Table 1.3.

(1) Multipart DSA Signature Generation without Simultaneous User Operations

DSA is a standard digital signature and promising because its scheme is applicable to elliptic curve encryption [KN1994]. However, it requires simultaneous user operations, when the private key is divided. Since the users, divided key holders, need to share a random secret and then calculate a signature with the random secret and private key, the users are required simultaneous broadcast communications to calculate the signature from two numbers.

Since the random secret can share before signature generation, the first number for the next signature can be calculated during the current signature generation, that is, the second number of the current signature. In this research, the interaction is processed via a server; the exchanged data between the key holders are put on the server in encrypted form. In order to show the solution is realistic, the performance of a prototype on a smartcard is evaluated. It is also proved the solution is as safe as the original DSA.

(2) Reducing Certificate Revocation and Certificate Verification Service with Time Stamp

The reason of signature invalidation is the revocation of signer's certificate after the signature verification. Therefore, it is sufficient to prove the validity of the certificate at

that time of the verification. Certificate Verification Service with Time Stamp (CVSTS) is the service to give the proof; the service gives the proof of the status of the public key certificate of the signer and existence of the signed document at a time point. The verifier can prove that the verification succeeded at that time. The scalability of the service should be evaluated.

Table 1.3: Research Strategies

Method	Technology	Problem	Solution
Protection by document itself	Private key protection by dividing	All key holders must sign simultaneously.	Multiparty DSA signature generation w/o simultaneous user operations
	Revocation of public key certificate with CRL	Digital signature becomes invalid with certificate revocation.	Reducing certificate revocation and certificate verification service with time stamp
	Encryption and signing of document part	Partial encryption and signing are not supported.	Partial encryption and signing for ODA document
Protection outside document	Access control with security attribute	Access control in multiple domains is insufficient.	Authorization with security attribute and privilege delegation
	Protection from unknown virus with system isolation	Multiple client use changes user operation.	Integration of network clients in system isolation with gateway

(3) Partial Encryption and Signing of ODA Document

The Open Document Interchange Format (ODIF) of the Office Document Architecture (ODA) [ISO1988] is an international standard of structured office document format; the format contains profiles, objects and its classes such as chapter, section, page, etc., and content portions. The ODA Security Addendum [ISO1990] defines encryption and signing method of ODIF, and realized partial encryption and signing. However, the standard is inconsistent with the PKI standard [CCITT1988] and there is a problem during editing process. In the research, solutions to the inconsistency are proposed.

(4) Access Control with Security Attribute and Privilege Delegation

While a client computer is basically used by a single user, a server computer is accessed by multiple users, and user authentication and authorization to resources are inevitable. In the research, firstly authorization requirements for document servers are defined, and security attributes used for access control are categorized. Secondly privilege delegation across security domain boundary is discussed.

(5) Integration of Network Clients in System Isolation with Gateway

In order to make system secure fundamentally, system isolation technology is adopted; multiple workstations are used according to information categories, and the workstations are integrated into a single PC with virtual machine and secure OS. The user of an existing such system has to be conscious of multiple categories, but a normal user, for example those of a commercial company, does not have such awareness. In the research, information is categorized into two, one is safe secret and the other is unsafe non-secret which may contain virus, and secure integration of e-mail clients of the different categories is realized with gateways with one-way information flow property.

1.4 Outline of the Dissertation

The reminder of this dissertation is organized as follows.

Chapter 2 proposes a multiparty DSA signature generation without simultaneous key holders' operations [SY2001, SH2001, SY2004]. The generation scheme has the following properties: (1) valid signatures are generated with odd n split private keys, (2) broadcast messages between the key holders are hidden from them, so that the n key holders do not need to process signature generation simultaneously, and (3) even if up to $t\,(=(n-1)/2)$ split keys are stolen, the adversary can get no information on the private key. Performance evaluation of prototype on smartcard and security consideration are described as well.

Chapter 3 proposes the attribute with validity period which reduces certificate revocation and CVSTS [SY1997a]. After the problem of public key certificate revocation is described, the attribute and service are defined. Next, performance of the CVSTS is evaluated followed by security evaluation.

In Chapter 4, firstly requirements of document interchange are discussed. Next ODA and its security are described as well as researches of security of ODA document as a whole. Then compatibility problems with the PKI standard and resolutions are discussed in details followed by problems of combination with existing ODA editor

[SY1995, SY1996].

Chapter 5 treats problems of access control or authorization of access to documents on a server [SY1997c]. Firstly authorization requirements of a document server are defined followed by the problems of the ACL. Next, an authorization scheme with the combination of a PAC and a Control Attribute Package (CAP) is proposed and how it matches the requirements is discussed. Moreover, problems of privilege delegation across domains and solutions with the PAC are discussed as well as a prototype on WISA server.

In Chapter 6, as a protection system of secret document leakage and virus infection, 'Windows Vault,' an integrated system of two Windows workstations [SY2005, SY2007] is introduced. Firstly, its concept is described followed by problems of an existing system. Next, the architecture including gateways connecting the two workstations securely is described, and performance of a prototype is evaluated. Security of Windows Vault is considered in detail as well as discussion on enhancement of the gateways and usability of network applications.

Finally, Chapter 7 concludes the results from this research and shows directions for the future research.

Chapter 2

Multiparty DSA Signature Generation without Simultaneous User Operations

2.1 Introduction

In this chapter, Multiparty Signature Generation (MSG) of DSA without simultaneous operation [SY2001, SH2001, SY2004] is described.

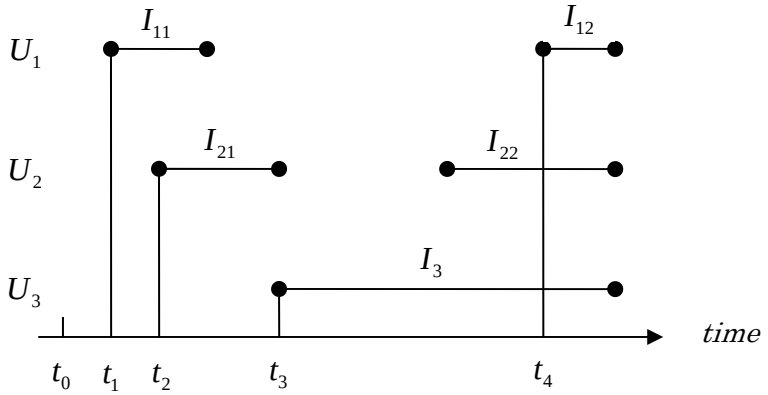
A typical method to protect signature key is to use smartcard, but the smartcard cannot prevent a malicious key holder from abuse. MSG is a technology to protect signature key from such threat; the signature key is divided into multiple pieces which are hold by multiple key holders, and the key holders cooperate to make a signature without revealing the divided keys to the other key holders. The multiparty RSA signature generation is straightforward, but the MSGs of DSA [CM1993, GR1996] are complicated and less realistic; the signature generation process requires broadcast messages between the key holders, and the key holders must process the MSG at the same time. It is out of touch with reality that all the key holders gather for each time of signature generation. Considering use in real world, it is much better that each key holder executes the signature operation at her/his convenience.

This chapter presents an MSG scheme of DSA without simultaneous operations of the key holders; the broadcast messages are hidden from the key holders, so that they do not need to execute signature generation at the same time. The key idea is simple and obvious, however, it brings great convenience. The key holders calculate signature parameters before the actual signature generation process, that is, during signature generation, each key holder calculates secret and public shares used for future signature generations. The data required to compute the shares are exchanged via a server; each key holder calculates temporary data, puts them on the server, retrieves them from the server, and these processes are repeated until the key holder obtains the shares required to generate the signature.

The benefit of the new scheme is illustrated in Figure 2.1. In this example, key holders U_1, U_2 and U_3 are in office, and can execute the signature generation operations during I_{11} and I_{12}, I_{21} and I_{22}, and I_3 respectively. Even if all the key

holders wish to sign a message at t_0, the signature cannot be generated in the existing schemes until t_4 when all the key holders are in office and execute the signature generation operations simultaneously. In contrast to the existing schemes, U_1, U_2 and U_3 execute the signature generation operations at t_1, t_2 and t_3 respectively, and the signature is generated at t_3 in the proposed scheme.

After the approach to realize the new MSG of DSA is described in Section 2.2, preliminaries of the original DSA and the verifiable secret sharing are described in Section 2.3, the MSG is presented in Section 2.4 and its performance is estimated in Section 2.5. The security of the MSG is described in Section 2.6 and the extension to threshold signature in Section 2.7. The application to other schemes is presented in Section 2.8, related works in Section 2.9 and conclusions in Section 2.10.



A signature is generated at t_3 earlier than t_4 when all the key holders gather.

Figure 2.1: Benefit of Presented MSG

2.2 Approach to New MSG of DSA

This section describes outline of the new MSG of DSA. The DSA signature consists of two numbers and looks like as follows:

$$(r, s) = (g^c, (message + ra)/c),$$

where g is a fix number, a is the signature key, and c is a random number which is generated every time of signing to $message$. In the MSG, the key and the random number are divided as follows:

$a = a_X + a_Y + a_Z$ and

$c = c_X + c_Y + c_Z$

where a_X, a_Y, and a_Z are the divided keys of key holders X, Y, and Z

respectively, and so on. The MSG calculates signature of the same form:

$$(r, s) = (g^{c_X + c_Y + c_Z}, (message + r(a_X + a_Y + a_Z))/(c_X + c_Y + c_Z))$$

without revealing secrets, a_X, a_Y, a_Z, c_X, c_Y, and c_Z, to the other key holders; the key holders exchange public and secret shares of each secret, and calculate parts of the signature from secrets.

The first number of the signature, $r = g^{c_X + c_Y + c_Z}$, can be calculated without message or document, so the key holders calculate the number before the actual signature generation process; during signature generation, each key holder calculates the secret and public shares used for future signature generations. The data required to compute the shares are exchanged via a server; each key holder calculates temporary data, puts them on the server, retrieves them from the server, and these processes are repeated until the key holder obtains the shares required to generate signature.

However, the exchange has two conditions that are required from security; firstly the key holders should have the same public shares of each key holder, secondary the shares should be exchanged at the same time. In order to satisfy the conditions, each key holder sends commitment of the public shares via the server before the key holder sends the public and secret shares; Figure 2.2 illustrates the share exchange of X mainly.

In the generation step, each key holder calculates its shares *Share* and commitment *Commit*, and sends *Commit* to the server. At the end of this step, the commitments of all key holders are stored on the server. Next in the distribution step, each key holder retrieves the commitments of the other key holders, and sends its share *Share*. At the end of this step, the shares of all key holders are stored on the server. In the verification step, each key holder retrieves *Share*'s of the other key holders and verifies that the commitment is calculated from the share of each of the other key holders. If the verification succeeds, each key holder calculates the hash value of all the public shares of all the key holders, *HashAll*, and sends it to the server. At the end of this step all *HashAll*'s are stored in the server. Finally in the production step, each key holder retrieves the other's *HashAll*'s and verifies that all of them are the same. Through the four steps, the public and secret shares are exchanged satisfying two conditions. The first condition, all share should be the same, is confirmed in the production step by checking that all *HashAll*'s are the same. The second condition,

shares should be exchanged at the same time, is satisfied by sending $Commit$ before $Share$; no key holder can change his share before receiving other's share. The details of exchange are described in Section 2.4.2.

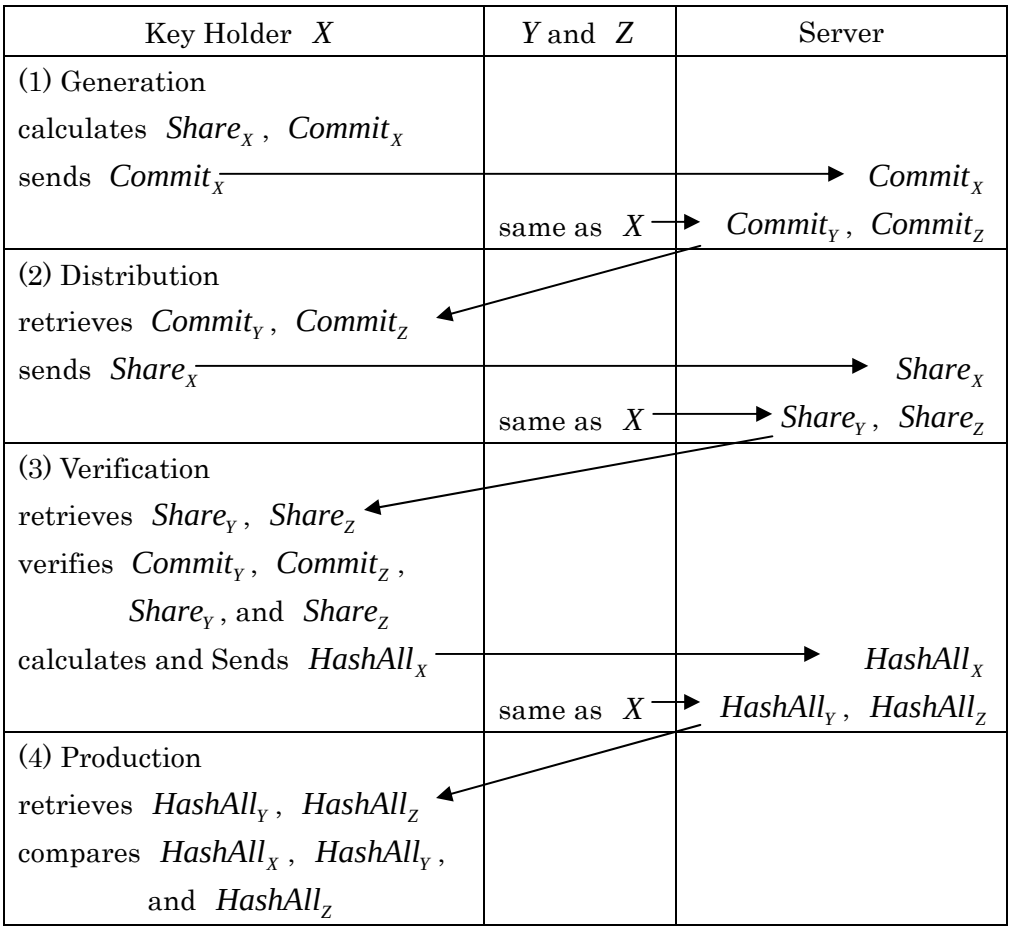


Figure 2.2: Share Exchange between Key Holders via Server

It is assumed that the number of the split keys is small, because the original private key is very important and it is difficult to consider such critical key is hold by many key holders even if it is split.

2.3 Preliminaries

2.3.1 Notations

Table 2.1 describes the notations used in the paper. The x-th power of F over Z_p is

written as $\exp(F, x)$; usually it is written as $F^x \mod p$, but the former notation is used, because there are many calculations in the power part such as

$$F^{\sum_{i=1}^{n}\prod_{j\neq i}\frac{j}{j-i}a_j \bmod q} \bmod p$$

and the following notation is easier to read

$$\exp(F, \sum_{i=1}^{n}\prod_{j\neq i}\frac{j}{j-i}a_j \bmod q).$$

Table 2.1: Notations

Notaition	Description
U_i	Key holder indexed with $i \in \{1, \cdots, n\}$
p	Large prime number
q	Large prime number dividing $p-1$
Z_p	Finite field of order p
Z_q	Finite field of order q
G	Element of Z_p of order q
$\exp(F, x)$	x-th power of F over Z_p
$mssg$	Message to be signed
$hash$	Cryptographic hash function whose range is Z_q
$\{mssg\}_{E(i,j)}$	$mssg$ encrypted with key shared by U_i and U_j
$\{mssg\}_{S(i,j)}$	Keyed hash value of $mssg$

It is assumed that each pair of the key holders shares a secret key of a

symmetric key encryption algorithm. The notation $\{mssg\}_{E(i,j)}$ represents the encrypted data generated by U_i with the following properties:

- Only U_i and U_j can access to $mssg$.
- U_j can verify that $mssg$ is originated from U_i.

U_i can generate such data by encryption of the concatenation of $mssg$, sender identifier i, receiver identifier j and the hash value of $mssg$, with the shared key. The notation $\{mssg\}_{S(i,j)}$ represents the keyed hash value of $mssg$, that is, the hash value of the concatenation of $mssg$ and the shared key. U_j can confirm $mssg$ is originated from U_i by checking the equality of the received $\{mssg\}_{S(i,j)}$ and the one generated by U_j itself. The integrity of $mssg$ is also checked.

2.3.2 The Digital Signature Algorithm

In this subsection, the DSA [NIST1998] is described. The DSA specifies the Secure Hash Algorithm One (SHA-1) [NIST1995] as the hash function, and it is written as $hash()$ simply.

(1) Key Generation

A key holder chooses randomly $a \in Z_q$ and calculates $P = \exp(G, a) \in Z_p$. The private key of the key holder is a, and the public key is the tuple of p, q, G and P.

(2) Signature Generation

For each signature generation, the key holder generates randomly non-zero number, $c \in Z_q$ and calculates the signature (r, s) as follows:

$$r = \exp(G, c^{-1} \bmod q) \bmod q,$$
$$s = (hash(mssg) + ra)c \bmod q.$$

Note that the random number c and the first part of the signature r are independent to the signed message $mssg$, and can be generated and calculated before the message is given [SC1989]. The presented scheme utilizes this property; the key holders calculate c and r before message is given, and s is calculated with pre-computed c and r.

(3) Signature Verification

If the following equation holds, then (r, s) is a valid signature of the message $mssg$:

$$r = (\exp(G, hash(mssg)\, s^{-1}) \times \exp(P, rs^{-1})) \bmod q.$$

2.3.3 Verifiable Secret Sharing

The (t, n, M) threshold verifiable secret sharing $(t+1 \le n \le M)$ is a scheme that a dealer distributes shares of a secret s to M key holders $U_1, \cdots, U_M$ with the following properties:

- Any group of at least n key holders can reconstruct s in polynomial time.
- Each key holder can verify that his share distributed by the dealer is correct, that is, after verification, all the key holders can make sure that any n key holders can reconstruct the correct secret.
- Any corrupted key holders up to t cannot get any information of s.

Pedersen showed a $(t, t+1, M)$ threshold verifiable secret sharing scheme [PT1991a, PT1991b] described in Appendix A.

2.4 MSG of DSA

In this section, the MSG scheme of the DSA is described. The following is assumed to realize the scheme:

- The scheme consists of key holders U_i and a server S.
- Up to $t \ge 1$ key holders and S may be corrupted. The corrupted key holders and S may do the eavesdropping, halting or malicious attacks [GR1996]. The scheme

prevents the private key from the eavesdropping attack, however, the key and signature generation procedures stop in the case of the halting and malicious attacks.

- The number of the key holders is $n = 2t + 1 \geq 3$. Extension to the threshold signature, where the private key is split into M keys such that $M \geq n + 1$, is described in Section 2.6.
- The communication channel may not be secure except during the execution of the first three steps of the key generation procedure described in the next subsection.

2.4.1 Key Generation

During the first three steps, S and the communication channel are assumed to be secure. This condition is required to keep the authenticity of the initial exchange of keys used in the later procedures. After the three steps, each pair of two key holders shares a key used for secure communication. In the following the key holders are indexed with i or j in $\{1, \cdots, n\}$.

(1) U_i chooses a random polynomial of degree t over Z_q:

$$h_i(x) = h_{i,0} + h_{i,1}x + \cdots + h_{i,t}x^t,$$

where $h_{i,0,}, \cdots, h_{i,t} \in Z_q$ and $h_{i,t} \neq 0$. U_i calculates the secret shares $a_{i,j}$'s, public shares $A_{i,m}$'s and commitment of the public shares A_i:

- $a_{i,j} = h_i(j) \bmod q (j \neq i),$
- $A_{i,m} = \exp(G, h_{i,m}) \bmod p \ (0 \leq m \leq t),$
- $A_i = hash(A_{i,0}, \cdots, A_{i,t})$,

and sends A_i to S.

(2) After all the key holders finish the above step, U_i retrieves $A_j (j \neq i)$ from S, and then sends $A_{i,m} (0 \leq m \leq t)$ to S.

(3) After all the key holders finish the above step, U_i retrieves $A_{j,m}(0 \le m \le t, j \ne i)$ from S, and verifies the following:

$$A_j = hash(A_{j,0}, \cdots, A_{j,t}) \ (j \ne i).$$

If the verification fails, then the key generation procedure stops and U_i quits from the procedure. This leads to the stop of the procedure executed by the other key holders. Otherwise U_i calculates $K_{i,j} = \exp(A_{j,0}, h_{i,0})$. $K_{i,j} = \exp(G, h_{1,0}h_{j,0}) = K_{j,i}$ is the shared secret key between U_i and U_j [DW1976]. U_i puts the following on S:

- $\{A_i'\}_{S(i,j)}$ where $A_i' = \{A_{1,0}, \cdots, A_{n,t}\}$,
- $\{a_{i,j}\}_{E(i,j)}$.

(4) After all the key holders finish the above step, U_i retrieves $\{A_j'\}_{S(j,i)}$ and $\{a_{j,i}\}_{E(j,i)}$ from S $(j \ne i)$, and verifies the following for each $j \ne i$:

- $\exp(G, a_{j,i}) = (\prod_{m=0}^{t} \exp(A_{j,m}, i^m)) \bmod p,$
- $\{A_j'\}_{S(j,i)}$ is consistent with A_i'.

If the verification fails, then the key generation procedure stops.

The public key of the key holders is (p, q, G, P) where P is calculated as follows:

$$P \equiv \prod_{i=1}^{n} A_{i,0} \equiv \prod_{i=1}^{n} \exp(G, h_{i,0}) \equiv \exp(G, \sum_{i=1}^{n} h_{i,0}) \pmod{p}.$$

2.4.2 Random Sharing

The first part of signature r is independent to the message to be signed, and can be calculated through the random sharing procedures before the actual message is given. This is the same idea of the preprocessing of the random number exponentiation [SC1989]. The procedure consists of four procedures, that is, generation, distribution,

verification and production procedures described in the following clauses.

During the procedures, the key holders generate random numbers (polynomials), distribute their secret and public shares in verifiable form using the verifiable secret sharing. In the end of the procedures, the key holders share secret $c \in Z_q$ and can generate $r = \exp(G, c^{-1}) \bmod q$, which correspond to c and r respectively described in Clause 2.2.2.2. When a message is given, the key holders can calculate the other part s of the signature with the secret shares of c and the private key through the signature generation procedure described in the next subsection. Each of the procedures must be executed for each signature.

The outline of the four procedures and the signature generation of the l-th message is described in the following.

(1) Random Generation Procedure

U_i generates two random polynomials of t degree $b_i^{(l)}$ and $c_i^{(l)}$, two random polynomials of $2t$ degree $v_i^{(l)}$ and $w_i^{(l)}$ whose constant terms are zero, and calculates the secret shares $b_{i,j}^{(l)}$, $c_{i,j}^{(l)}$, $v_{i,j}^{(l)}$, and $w_{i,j}^{(l)}$ for U_j. U_i also computes the public shares $B_{i,m}^{(l)}$, $C_{i,m}^{(l)}$, $V_{i,m}^{(l)}$, and $W_{i,m}^{(l)}$ shared by all the key holders, and sends the commitment to S:

- $\left\{CMT_i^{(l)}\right\}_{S(i,j)}$ where $CMT_i^{(l)} = \left\{B_{i,0}^{(l)}, \cdots B_{i,t}^{(l)}, C_{i,0}^{(l)}, \cdots, C_{i,t}^{(l)}, V_{i,1}^{(l)}, \cdots, V_{i,2t}^{(l)}, W_{i,1}^{(l)}, \cdots, W_{i,2t}^{(l)}\right\}$.

(2) Random Distribution Procedure

U_i retrieves $\left\{CMT_j^{(l)}\right\}_{S(j,i)}$ $(j \neq i)$ and then sends the encrypted secret shares and the public shares to S:

- $\left\{b_{i,j}^{(l)}, c_{i,j}^{(l)}, v_{i,j}^{(l)}, w_{i,j}^{(l)}\right\}_{E(i,j)}$ and
- $B_{i,m}^{(l)}, C_{i,m}^{(l)}, V_{i,m}^{(l)}, W_{i,m}^{(l)}$.

(3) Random Verification Procedure

U_i retrieves the secret and public shares, and verifies that $\left\{CMT_j^{(l)}\right\}_{S(j,i)}$ and $b_{j,i}^{(l)}$, $c_{j,i}^{(l)}$, $v_{j,i}^{(l)}$, $w_{j,i}^{(l)}$, $B_{j,m}^{(l)}$, $C_{j,m}^{(l)}$, $V_{j,m}^{(l)}$, $W_{j,m}^{(l)}$ are consistent. After the verification, U_i sends the following:

- $\left\{BCVW_i^{(l)}\right\}_{S(i,j)}$ where $BCVW_i^{(l)} = \left\{B_{1,0}^{(l)},\cdots,B_{n,t}^{(l)},C_{1,0}^{(l)},\cdots,C_{n,t}^{(l)},V_{1,1}^{(l)},\cdots,V_{n,2t}^{(l)},W_{1,1}^{(l)},\cdots,W_{n,2t}^{(l)}\right\}$.

(4) Production Procedure

U_i retrieves $\left\{BCVW_j^{(l)}\right\}_{S(j,i)}$ $(j \neq i)$ and verifies they are the same as the one generated by U_i itself. U_i can confirm that all the key holders have the same public shares with this verification. After the verification, U_i sends the production of two secrets with random number:

- $d_i^{(l)} = ((\sum_{j=1}^{n} b_{j,i}^{(l)})(\sum_{j=1}^{n} c_{j,i}^{(l)}) + \sum_{j=1}^{n} v_{j,i}^{(l)}) \bmod q$.

(5) Start-up Random Sharing

After the key generation procedure, the key holders execute firstly the random generation procedures of the first through forth signature generations $(l = 1,2,3,4)$, secondly the random distribution procedures of the first through third signature generations $(l = 1,2,3)$, thirdly the random verification procedures of the first and second signature generations $(l = 1,2)$, and finally the random production procedure of the first signature generation $(l = 1)$. The start-up procedure consists of the above procedures, and is executed only once after the key generation procedure. This is the preparation for the signatures of the first through forth messages to be signed. At the end of the start-up procedure, the key holders have the secret and public shares, and they can compute the first signature $(l = 1)$ with the shares. Table 2.2 illustrates the relation of the sub-procedures and the signature generation procedure.

RG	S	S	S	S	L1	L2
RD	S	S	S	L1	L2	…
RV	S	S	L1	L2	…	
RP	S	L1	L2	…		
SG	L1	L2	…			

RG: Random Generation
RD: Random Distribution
RV: Random Verification
RP: Random Production
SG: Signature Generation

S: executed as the start-up procedure
L1: executed during the first SG ($l=1$)
L2: executed during the second SG ($l=2$)

The table shows the relation of the procedures; for example, the RG, RD, RV procedure of the second signature ($l=2$) are processed as the start-up procedure (S), and the RP procedure during the first SG(L1)

2.4.3 Signature Generation

The following describes the l-th signature generation procedure. As the first step of the procedure, the l-th message $mssg^{(l)}$ is put on S.

(1) U_i retrieves $d_j^{(l)}$ $(j \neq i)$ and $mssg^{(l)}$ from S. After U_i confirms to sign $mssg^{(l)}$, U_i calculates the following and sends them to S:

- $r_i^{(l)} = \exp(\prod_{j=1}^{n} B_{j,0}^{(l)}, (\sum_{j=1}^{n} \prod_{k \neq j} \frac{k}{k-j} d_j^{(l)})^{-1} \bmod q) \bmod q$ and
- $s_i^{(l)} = ((hash(mssg^{(l)}) + r_i^{(l)} \sum_{j=1}^{n} a_{j,i})(\sum_{j=1}^{n} c_{j,i}^{(l)}) + \sum_{j=1}^{n} w_{j,i}^{(l)}) \bmod q$.

(2) U_i executes the random generation sub-procedure of the $(l+4)$-th signature,
(3) executes the random distribution sub-procedure of the $(l+3)$-th signature,
(4) executes the random verification sub-procedure of the $(l+2)$-th signature, and
(5) executes the random production sub-procedure of the $(l+1)$-th signature.

After all the key holders execute the above procedures, S verifies that all $r_j^{(l)}$'s are the same and that the following final signature $(r^{(l)}, s^{(l)})$ is a valid signature of $mssg^{(l)}$:

$$(r^{(l)}, s^{(l)} = (r_1^{(l)}, (\sum_{i=1}^{n} \prod_{j \neq i} \frac{j}{j-i} s_i^{(l)}) \bmod q).$$

If the verification fails, the signature generation procedure stops. Otherwise S outputs the signature. At this stage the key holders have the public and secret shares that are required to compute the $(l+1)$-th signature. For example, when the first signature is generated $(l=1)$, then the key holders have the shares of the second signature $(l=2)$. The relation of the sub-procedures is illustrated in Table 2.

2.5 Performance Estimation

A prototype of the MSG scheme has been implemented on a smart card and the performance is estimated in the environment shown in Table 2.3.

Table 2.3: Environment of Performance Estimation

item	description
t	1
n	3
CPU	SLE66CX160S
Memory size	16K bytes
OS	MULTOS v4.0
Length of p	1,024 bits
Length of q	160 bits
Hash algorithm	SHA-1
Encryption algorithm	Triple DES

Table 2.4: Performance

Step	Procedure	Time (seconds)
1	Generation of (r_i, s_i)	1.7
2	Random Generation	11.4
3	Random Distribution	3.0
4	Random Verification	41.1
5	Random Production	0.6
N/A	Total	57.8

Some processes are different from the description in Section 2.3 in order to save the memory space. For example, $b_{i,j}^{(l)}$, $c_{i,j}^{(l)}$, $v_{i,j}^{(l)}$, and $w_{i,j}^{(l)}$ are encrypted during the parameter generation sub-procedure, not the parameter distribution sub-procedure. The values of $\prod_{j=1}^{n} B_{j,0}^{(l)}$ and $\sum_{j=1}^{n} c_{j,i}^{(l)}$ are calculated during the parameter verification sub-procedure.

The performance of the prototype is shown in Table 2.4. The total time of the signature generation is about 58 seconds, and it cannot say that it is sufficiently fast as interactive use with human user. However, the scheme can be used in interactive way with the following changes:

- The signature generation procedure is divided into two parts, the generation of $(r_i^{(l)}, s_i^{(l)})$ and the parameter sharing and the later is processed as background. As a result, it looks for the human user that the signature procedure finishes in 1.7 seconds, the processing time of the generation of $(r_i^{(l)}, s_i^{(l)})$, and this is sufficient performance for interactive use with human user.
- The users might want to sign multiple k messages at a time. In order to realize this, it is required that each of the random sharing procedures should process multiple randoms; the random production procedure generates $d_i^{(l+1)}, \cdots, CMT_j^{(l+2k)}$, instead of just $d_i^{(l+1)}$, and the random verification procedure checks

$CMT_j^{(l+k+1)}, \cdots, CMT_j^{(l+2k)}$ instead of just $CMT_j^{(l+2)}$, and so on.

2.6 Security of MSG

In this section, the compatibility and security of the presented scheme are presented; it is proved that the presented scheme is compatible with and as secure as the original DSA. We also discuss the security of the server that may be corrupted.

2.6.1 Compatibility with Original DSA

The following theorem shows that signatures generated with the MSG are verified with the same way as the original DSA.

Theorem 1 (Compatibility with the original DSA): The signature $(r^{(l)}, s^{(l)})$ is a valid DSA signature of the message $mssg^{(l)}$.

The proof of the theorem is given in Appendix B.

2.6.2 Unforgeability of MSG

It is shown that the MSG is as secure as the original DSA against the adaptive chosen message attack with the following assumption: the original DSA is secure against the attack even if adversary knows the first part of signature r previously, that is, the adversary can choose message with knowledge of the part. It is considered that this assumption does not give impact on the security because of the following reasons:

- r is generated randomly and out of control of the adversary as well as the honest key holders, and
- the signature is calculated from the value of $hash(mssg)$, not $mssg$, and it is difficult to choose adequate $mssg$ for the attack.

In case of the attack against the MSG, the concept of view is required; the view of U_1 is everything that U_1 sees during the execution of the key generation, parameter sharing and signature generation procedures. Example of the view of U_1 is given in Appendix C.

An adversary X for the original DSA is allowed to use a key holder as an oracle; X tries to forge a signature of a message $mssg$ after it gets signatures of messages of its own choice. The oracle gives the first part of the signature r before X requests a signature. The oracle is different from one described in [PC1996]. The notation $(mssg')$ is used to denote the chosen messages $mssg_1', mssg_2', \cdots$. If there is no such probabilistic polynomial time algorithm for X, then it is called that the DSA is secure against the adaptive chosen message attack. The notation $X(p,q,G,P)$ is used to denote the random variable that takes a value of $((mssg'),(mssg,r,s))$ with the same probability that X queries $(mssg')$ to the oracle and finally outputs $(mssg,r,s)$ on input $X(p,q,G,P)$.

An adversary Y for the MSG of the DSA that corrupts up to t key holders is allowed to use n key holders as an oracle; Y tries to forge a signature of the target message $mssg$ with signatures of messages of its own choice $(mssg')$ got from the n key holders including the t corrupted key holders. The view of Y is the sum of the ones of the corrupted key holders. If there is no such probabilistic polynomial time algorithm for Y, then it is called that the MSG of the DSA is secure against the adaptive chosen message attack. The notation $Y(p,q,G|P)$ is used to denote the random variable that takes a value of $((mssg'),(mssg,r,s))$ with the same probability that Y queries $(mssg')$ to the oracle and finally outputs $(mssg,r,s)$ under the condition that the generated public key is (p,q,G,P). The following theorem shows that the MSG is as secure as the original DSA against the adaptive chosen message attack.

Theorem2 (Unforgeability): For any adversary Y for the MSG of the DSA, there is an adversary X for the original DSA such that

$$\Pr(X(p,q,G,P)=((mssg'),(mssg,r,s))) = \Pr(Y(p,q,G|P)=((mssg'),(mssg,r,s)))$$

for any public key (p,q,G,P) and any $((mssg'),(mssg,r,s))$.

The proof of the theorem is given in Appendix D.

2.6.3 Server Failure

It is clear that S is the weak point of the MSG, but it is considered that the failure of the server is not a severe problem. If the server is unavailable, the key holders cannot

generate signatures, however, the failure of the server does not affect the security of the scheme; even if the server is corrupted, the adversary cannot get any information of secret keys of the key holders, nor generate valid signatures. This security level is the same as the one against the eavesdropping adversary of the case that $n \geq 2t+1$ [GR1996].

The problem is that the key holders cannot generate valid signatures because of the disturbance of the server, a sort of denial of service (DoS) attack. This attack is considered to be difficult to prevent completely; an adversary can also do the attack by sending plenty of bogus data to the network connecting the key holders.

A counter measure against the attack is to protect the server and network from such attack. This is the same availability as the other existing MSG schemes assuming secure channel between the key holders. If the network between the key holders of the existing MSGs is under the DoS attack, they cannot generate signature, because they cannot exchange shares.

2.7 Extension to Threshold Signature

In this section, the presented MSG scheme is extended to the threshold signature scheme. Firstly the use of the extended scheme is described. Usually n out of M ($M \geq n+1$) cards (key holders) are used to generate signatures and the other $M-n$ cards are kept in safe as spare cards. If one of the n cards becomes unavailable, for example the card is broken or lost, then one of the $M-n$ spare cards is used instead of the unavailable card. Even if t cards are lost, the system is secure under the condition that there is no adversary in the n card holders. When a spare card is newly used, the start-up parameter sharing procedure is executed with the new combination of the n cards.

The extension is applied basically to the key generation procedure; each U_i generates h_i and distributes $a_{i,j} = h_i(j) \bmod q$ to $U_j (1 \leq j \leq M)$. The random sharing and signature generation procedures are slightly different; the summations and products are calculated with the indices of the chosen n cards. For example, if $U_2, \cdots, U_{n+1}$, are used, then the final signature s is calculated as follows:

$$s = \left(\sum_{i=2}^{n+1} \left(\prod_{j=2}^{j=i-1} \frac{j}{j-1}\right)\left(\prod_{j=i+1}^{n+1} \frac{j}{j-1}\right) s_i\right) \bmod q .$$

2.8 MSG of Other Signature Schemes

The MSG scheme can be applied to the other signature schemes based on the DLP, such as the Nyberg-Rueppel Signature (NRS) [IEEE1999]. The key holder chooses a random number $c \in Z_q$ and calculates a signature (r, s) of the NRS for the message $mssg$ as follows:

$$\begin{aligned} r &= (\exp(G, c) + hash(mssg)) \bmod q, \\ s &= (c - ra) \bmod q \end{aligned}$$

where $a \in Z_q$ is the private key.

The basic idea of the MSG of the NRS is exactly the same; the parameter r is calculated before the actual signature generation. Key holder U_i processes $b_{i,j}$ and $B_{i,m}$ in the same way as in Section 2.3.2 where $t = n - 1$, and then outputs the following:

$$\begin{aligned} r &= (\prod_{j=1}^{n} B_{j,0} + hash(mssg)) \bmod q, \\ s_i &= (\sum_{j=1}^{n} b_{j,i} - r \sum_{j=1}^{n} a_{j,i}) \bmod q, \end{aligned}$$

and then the server outputs the final signature:

$$(r, s) = (r, \sum_{i=1}^{n} \prod_{j \neq i} \frac{j}{j - i} s_i \bmod q).$$

Comparing with the MSG of the DSA, the MSG of the NRS is secure against the eavesdropping attack even if $n - 1$ key holders are corrupted, while the limit of the corrupted key holders is $t = (n - 1)/2$ in case of the DSA.

The idea can be also applied to the digital signatures based on the elliptic curve DLP [KN1994] such as the Elliptic Curve DSA, the Elliptic Curve NRS [IEEE1999]. This is because a part of signature can be generated without the message to be signed. The other part of the signature is calculated with the message, the pre-generated parameter and the split key without simultaneous operations of all the key holders.

2.9 Differences with Existing MSG

Many MSGs are presented in recent years. Comparing with the ones of the RSA cryptosystem [GN1999, MM1999], the presented MSG is more complicated and less efficient, because of the many interactions between the key holders and server. However, the presented system is more promising than the MSGs of RSA, because the MSG can be applied to the elliptic curve cryptosystem, which will be used more than RSA in near future.

Comparing with the ones based on the DLP [CM1993, GR1996, MK2001, PC1996], it is considered that the presented system has much advantage in the real world use, because the MSG does not require simultaneous operations of the key holders and each of them can sign at her/his convenience. The basic idea is simple and obvious, but it was out of consideration of the previous works. As for the most critical security, the presented system is as secure as the original DSA against the strongest attack, the adaptive chosen message attack, with the assumption that the original DSA is secure against the attack even if adversary knows the first parameter r previously. As for attacks by corrupted key holders, the MSG is as secure as the existing system against the eavesdropping attacks, even if the server is also corrupted. The server is the weak point, however, it is considered that the defect is supplemented by the advantage in the real world use. Note that the other schemes assume that the network between the key holders is secure, and its security is out of consideration.

The least number of key holders is three; $n = 2t + 1$ and $n = 3, 5, 7, \cdots$. There is a MSG of DSA for two holders which does not require simultaneous user operations [MP2001], but it needs much more computation; the MSG needs more than twenty exponential computation per key holder during signature generation, while the proposed MSG is one. As a result, it is not realistic to implement the MSG on smartcard.

The number of messages exchanged between the key holders is greater than the other MSGs, for example DSS-Thresh-Sig-1 [GR1996], because the presented system detects a malicious attack, and stops the key and signature generation procedures. This is also true as for the computational cost. However, it is shown that a signature is generated in 1.7 seconds in Section 2.5 and it is considered the performance is sufficient for interactive use with human user.

2.10 Conclusions

In this chapter the MSG of the DSA without simultaneous processing is presented; with the pre-computation of a part of signature, the broadcast messages are hidden from the key holders, and it is possible for each key holder to process the signature generation at her/his convenience. The security of the scheme, the performance estimation of a prototype on a smartcard, the extension to the threshold signature and the application of the MSG to other signature schemes are also discussed.

The MSG can realize secure key management that is easier to use than the existing MSGs. The MSG will be used widely from the view that the scheme can be directly applied to the elliptic curve cryptosystem.

The remaining problem is the flexibility on constitution of key holders; the order of key holders' processing is not fixed, but the constitution of key holders is fixed. It is convenient that any n key holders out of M sign a message, then a DSA signature of the message is produced.

Chapter 3

Certificate Verification Service with Time Stamp Solving Invalidation of Signature by Certificate Revocation

3.1 Introduction

This chapter describes a user attribute with validity period extension field of a Public Key Certificate (PKC) and a Certificate Verification Service with Time Stamp (CVSTS), which solve the problems of unavailability of the latest revoked certificate information, large size of the revocation information, and lack of non-repudiation mechanism of the PKI [CCITT1988].

A PKC binds a public key to its owner with her/his name optionally including affiliation with the signature of the issuer CA. A PKC user can get an authentic public key from the PKC, and verifies the signature of the PKC owner with the authentic public key. The PKC is just a static data and it may contain old information; the PKC is revoked when it is suspected that the private key was compromised or the affiliation of the owner changed, etc. The revocation is announced with a Certificate Revocation List (CRL) including the serial numbers of revoked PKCs.

However, the revocation announcement with the CRL has three problems; firstly unavailability of the latest revocation information, secondly large CRL size, and thirdly invalidation of signature with the revocation. These problems lead the following practical issues and disturb widespread use of the PKI; for example, transaction request which is actually invalid is verified as valid, or conversely contract document which was valid at signing time is verified as invalid later.

In order to solve these problems, a new extended field in the PKC called User Attribute with Validity Period (UAV) and a new on-line service, CVSTS are introduced. The targets of the two solutions are mainly the PKI applications which require the non-repudiation mechanism, such as secure messaging, digitally signed document, or business transaction.

This chapter is organized as follows: Section 3.2 describes the problems of the current revocation announcement method, and Sections 3.3 and 3.4 include the description of the new extended field, UAV, and the new service, CVSTS. In Section 3.5

the performance of the CVSTS is evaluated, and Sections 3.6 and 3.7 include discussions on the cost, security, scalability of the CVSTS, and related work. Finally Section 3.8 concludes this chapter.

3.2 Problems of Current Revocation Announcement

3.2.1 Unavailability of Latest Revocation Information

A CRL does not always contain the latest PKC status because the CRL is issued periodically, for example once a month, and a PKC user may get to know the revocation after the next CRL is issued. A simple solution to this problem is to issue the CRLs frequently and the user retrieves the latest CRL every time the user verifies a digital signature. But this solution is not feasible because of the problem described in the next subsection.

3.2.2 Large Size of CRL

The serial numbers of a revoked PKC are held in the CRL during the validity period of the PKC, and the CRL size may become large. Thus it may cost high for a user to retrieve the CRL through network and to hold it locally. This problem is serious in the case of an organizational CA; a major revocation reason of the PKC issued by the CA is the affiliation change of the owner; a large number of members of the organization change their departments or branches periodically, say April and October, and the owner names change. As a result, the number of revoked PKCs is large, and the size of the CRL increases. For example, if 10% of 10,000 employees of a company change their affiliations, the CRL size becomes about 20K bytes, and it is not feasible to retrieve such sizeable data each time of verification.

The version 2 CRL defined in the X.509 amendment [ISO1995] introduced two solutions, delta-CRL and CRL distribution point. The delta-CRL contains only difference from the previously issued CRL. The CRL distribution point is used to split the revocation information according to reasons of revocation, such as unuse, key compromise or suspension of key use, and to identify the location where the split CRL can be obtained. This helps to distribute the CRL repositories in network according to the revocation reason, and to reduce the cost of retrieval of the CRL when only revocation due to a specific reason is concerned.

With the help of the version 2 CRL, a CA can issue CRLs more frequently and

gives users more timely revocation information, because the transmission of the smaller CRL is lighter than the original CRL. However, a user machine that does not have secure storage for the revocation information needs to verify the base CRL and delta-CRL, and this makes the load of the user machine heavier. Therefore the version 2 CRL is not a complete solution against the CRL size problem.

3.2.3 Lack of Non-Repudiation Mechanism

With the timely PKC revocation information, a user that has machine with the secure storage for the revocation information can verify a digital signature at the time when the user receives it. This is adequate for verifying an origin of connection request or an update request of a database entry, because it is enough for such applications to authenticate the source of the request, and to detect unauthorized change of the application data. However, it is not enough for messaging service; after the revocation of the PKC containing the public key of the originator, the signature of the signed message becomes invalid illustrated in Figure 3.1, and there is no evidence that she/he has created the message, and the originator can repudiate the fact that the originator sent the message to the recipient or the content itself. Such applications are secure messaging, digitally signed document and business transaction data, etc.

In order to solve this problem the recipient needs the evidence of the fact that the message had existed or been received before the PKC was revoked. A Time Stamp Service (TSS) [HS1991] proves the fact; a TSS server, which is a Trusted Third Party (TTP), stamps the time with data requested by a user, and the time-stamped data is the evidence of the existence of the data at that time. The recipient can bring the CRL and time-stamped data to court for resolution of the dispute between the originator and recipient.

A recipient of a signed message firstly accesses a CRL repository to get the latest CRL and a TSS server to get the time stamp, and then keeps them for future use. However, there still remains the size problem of CRL, that is, the recipient whose machine does not have secure storage for revocation information needs to retrieve the sizeable CRL every time of verification. There is another size problem; in order to resolve a future dispute, the recipient needs to keep the large CRLs (base and delta-CRL) in order to prove the PKC is not revoked.

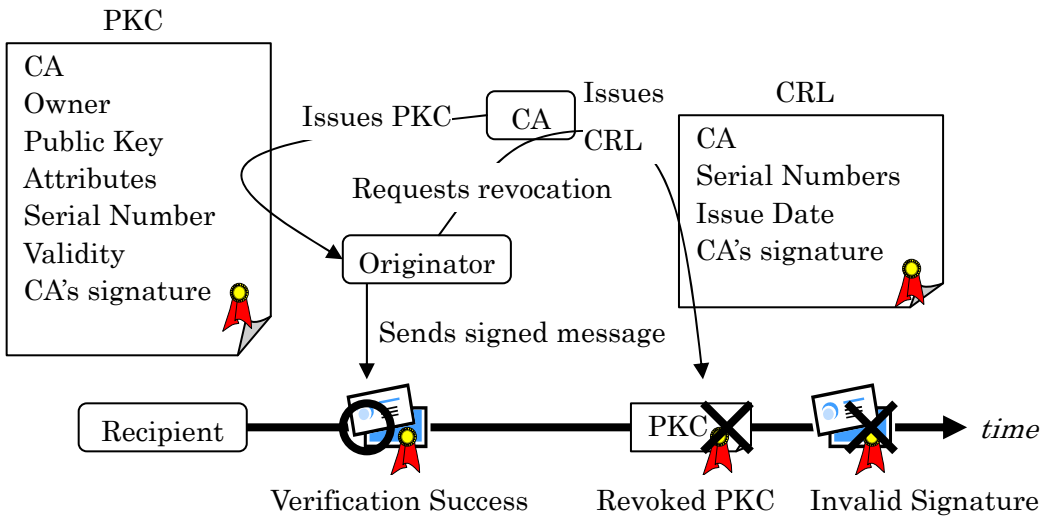


Figure 3.1: Invalidation of Signature with PKC Revocation

In the following, solutions to the problems are presented; the UAV is the solution to the large CRL size problem, and the CVSTS is a solution to the unavailability of the latest revocation information problem and the lack of non-repudiation mechanism problem. The combination of the two solutions resolves the problems caused from the current revocation mechanism of the PKI.

3.3 User Attribute with Validity Period Field

The UAV is introduced as a solution to the large CRL size problem, in particular in the case that the issuer CA is an organizational CA. The field contains user attributes, such as department name, title of the owner, with the validity period of the attributes. The owner name field does not contain the department and/or branch name of the owner. The following Abstract Syntax Notation One (ASN.1) [RT1990] description gives its syntactical definition:

```
UserAttributeWithValidityPeriod ::= SEQUENCE {
  notBefore          UTCTime,
  notAfter           UTCTime,
  userAttributes     SEQUENCE OF SEQUENCE {
    type               OBJECT IDENTIFIER,
    value              ANY defined by type } }
```

The userAttributes field is valid between notBefore and notAfter. Even if one of

the attributes becomes invalid after notAfter, the issuer CA does not revoke the PKC. Synchronizing the validity period of the UAV with the personnel changes can reduce the PKC revocation caused by the periodical personnel change. Figure 3.2 illustrates how to issue a PKC with the UAV in the case that there are periodical personnel changes every April and October. The validity period of the first PKC (a) is two years and the validity period of its UAV is half a year. If the owner changes her/his affiliation on 1999/10, the new affiliation information is stored in the UAV of the PKC (b) issued at the time. In this case, the key pair is also changed, and the validity period of the new PKC is two years. If the same key pair is required, the validity period of the PKC should be one and half a year (c), because the validity period depends on the lifetime of the key pair, and the end of it, 2001/4, specified in the old PKC (a), should not be changed. A new PKC is also issued to the member whose affiliation does not change, and the UAV of the PKC is the next personnel change time, because the validity of the UAV has expired.

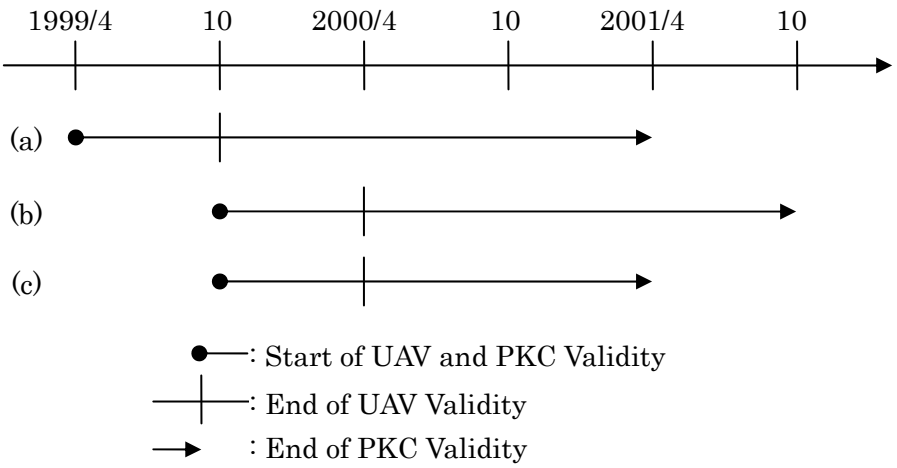


Figure 3.2: Validity Period of PKC and UAV

The simplest other solution without the UAV is to remove the department name from the owner name. However, this solution has a demerit and cannot be acceptable for digitally signed message of business use. Since the department name gives information on the role of the originator, the recipient can use it to judge the authenticity of the message, and therefore the removal deprives such chance from the recipient. As for the userDirectoryAttribute field [ISO1995] used to include the title/role of the owner, the situation is the same.

3.4 Certificate Verification Service with Time Stamp

3.4.1 Overall Architecture of CVSTS

While the UAV is a solution for the CRL large size problem, the CVSTS aims to resolve the unavailability of the latest revocation information problem; the CVSTS solves the invalidation of signature and the lack of non-repudiation mechanism problems. The CVSTS provides fresh PKC status and time stamp services, and the two services give the non-repudiation mechanism. The basic idea [SY1997a] is simple and almost the same as the Electronic Signature Timestamp Server (ESTS) [LJ1995]; a CVSTS client sends a request including identification information of certification path and the message digest of the data to be time-stamped, and then a CVSTS server sends back a signed response including the status of the PKC(s), the message digest and the current time. The following subsections describe the CVSTS architecture, the secure transmission of PKC information over network, which are not covered by the ESTS architecture, and the interaction between the CVSTS client and server.

Figure 3.3 describes the architecture of the CVSTS. A CA accepts a PKC issue or revocation request from a public key owner. After certification of the request, the CA updates the PKC information database holding of all the information on issued PKCs as well as revoked PKCs. A PKC information server announces the update of the database periodically, that is, the issued and revoked PKCs information, to a master CVSTS server. With help of the UAV, the size of the information can be reduced and the PKC information server can announce more frequently. The information also includes the update time.

The CVSTS server is operated by a CVSTS authority, which is a TTP. The master CVSTS server archives the update information and sends the information to the slave servers. Each of the slave servers has a local database and directly communicates with CVSTS user clients through network.

When the PKC users are quite many or scattered in large geographical region, then a hierarchical CA tree is constructed, and a high level CA receives PKC issue/revocation request from a low level CA. The high level CA announces the PKC revocation of the low level CA to the master CVSTS server as soon as the request arrives, because the revocation has much impact on the security infrastructure compared with that of end user.

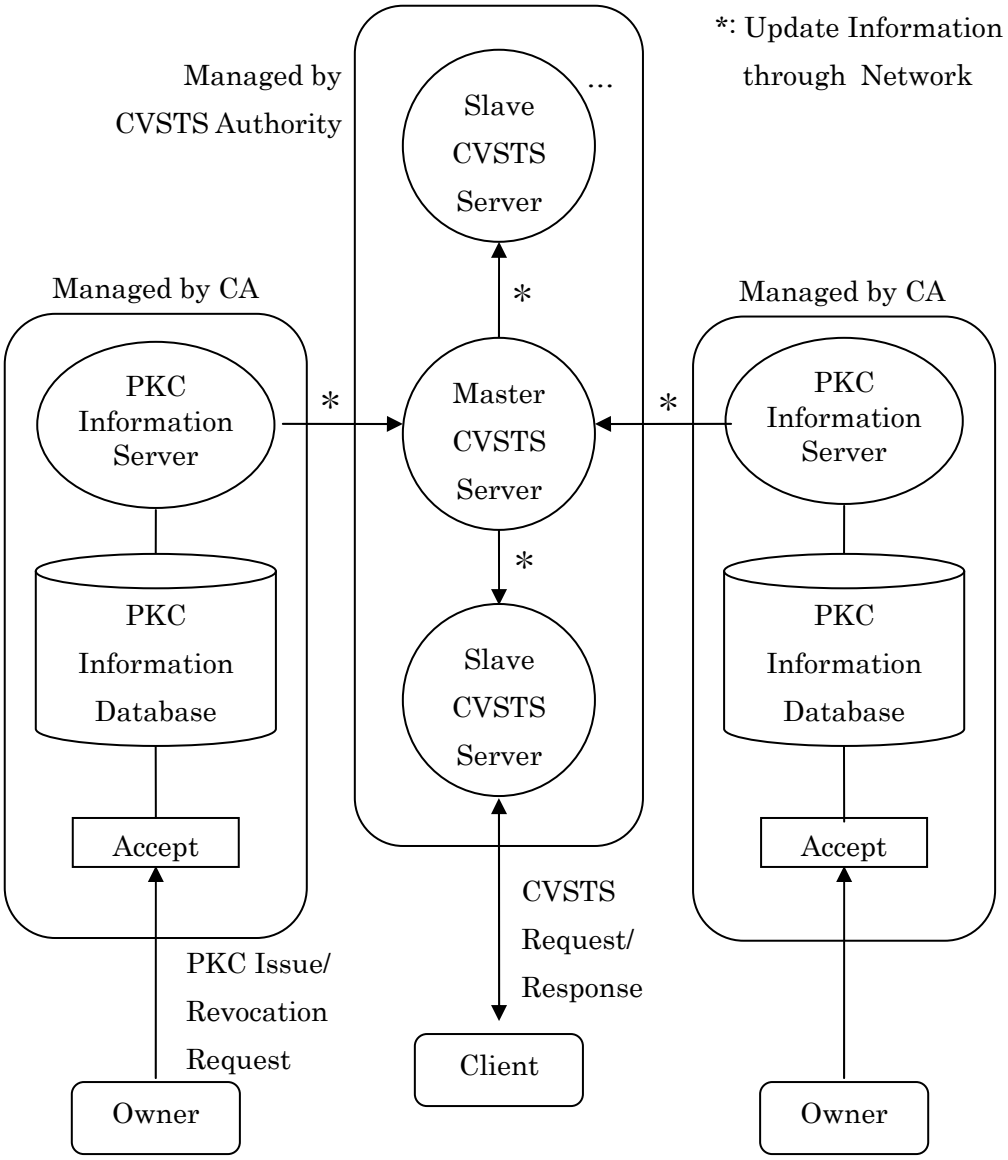


Figure 3.3: CVSTS Architecture

3.4.2 Secure Transmission of PKC Information between Servers

The update information should be propagated securely; in particular data origin authenticity and integrity are essential. The data enveloping technique, for example PKCS#7 digital envelope [RSA1993b], is used to achieve the security. It wraps application data and gives confidentiality, integrity and origin authenticity. Each of

enveloped update information is transmitted through an existing transport protocol.

The CVSTS server needs to know not only the revocation information, but also the validity period and the key usage time, because the client may request the status at a certain time point, for example the time when the signature of a message was generated, and the server checks the status with the validity period of the PKC and the keyUsage field as well as the revocation information.

The secure transport channel, such as the Transport Layer Security (TLS) protocol [DT1999], also realizes the data origin authenticity and integrity. However, the data enveloping technique is used with the following reasons:

(1) The signature of the CA or the PKC information server operated by the CA proves that the information is authentic.
(2) The update information will be used as evidence and audit trail in order to prove that the CVSTS authority operates the service properly according to the authentic information from the CA.

3.4.3 Interaction between CVSTS Client and Server

A user client accesses a slave CVSTS server; it sends a CVSTS request message containing identification information of a PKC or certification path of which status the user wants to know, and the message digest of the data that the user asks the server to time stamp. The server firstly checks the local database, secondly creates a message containing the status of the PKC or certification path, the message digest contained in the request and the current time, thirdly signs with the private key of the server, and finally sends the signed message, the CVSTS response message, to the client. The following clauses include the description of the request and response messages defined and encoded according to the ASN.1 and Basic Encoding Rule (BER) [RM1990]. If an error occurs, such as server internal error, then an error response is returned to the client.

(1) CVSTS Request

The ASN.1 definition of the CVSTS request message is given in Appendix E, and the meaning of each field of the message is described as follows:

- The field of version specifies the version of the message.

- The field of certificates field identifies the PKC(s) that the user wants to know the status.
- The field of verificationRequestTime contains the time when the user wants to know the PKC status. Normally the field is void, and the latest status is sent back from the server. However, when the user wants to know the status at a certain time point, then the field is set to the time.
- The field of dataToBeTimeStamped is a message digest of the data that the user wants the server to time stamp.
- The field of messageDigestAlgorithm identifies the algorithm used to generate the message digest.
- The field of additionalInformation contains a sequence of the following data and carries additional information of the data to be time-stamped:

  AdditionalInformation ::= SEQUENCE {
  type OBJECT IDENTIFIER,
  value ANY defined by type }

  The first field type specifies the type of the information and the second field value contains the associated value. Examples of the additional information are originator, creator, format, or title of the data.
- The field of requestOriginator is the name of the user that may be required to access the CVSTS server.
- The field of requestIdentifier is the identifier of the request.

The user optionally signs the message for the sake that the server can authenticate the requester.

(2) CVSTS Response

The ASN.1 definition of the CVSTS request message is given in Appendix E, and the meaning of each field is described as follows:

- The field of version specifies the version of the message.
- The field of issuer and serialNumber identify the PKC(s) of which the server has checked the status, and lastUpdate is the latest update time of the PKC information database of the issuer CA.
- The field of requestIdentifier is the identifier in the request.
- The field of verificationResult contains the status of the PKC(s) which values and

meanings are as follows:

```
VerificationResult ::= ENUMERATE {
    valid (0),
    notYetValid (1),
    finished (2),
    revoked (3),
    hold (4),
    unknown (5) }
```

If one of the PKCs is not valid, the PKC is specified in the invalidCertificate and its status is included in this field. The last value unknown means that the status is unknown with some reason such as the update information expected to arrive has not yet arrived at the server.

- The field of revokedReason contains the reason of the revocation. The possible reasons are the same as those of the amendment [ISO1995].
- The field of revokedOrHoldTime identifies the date and time when the PKC was revoked or held.
- The field of invalidTime specifies the date and time when the key was actually compromised, etc.
- The field of verificationTime specifies the date and time when the PKCs status is checked. This is the oldest time of the lastUpdate fields. If the request includes the verificationRequestTime and the server holds the PKC status at that time, the field is the same as the verificationRequestTime. If the server does not hold the status at the requested time, it returns a response according to the nearest but earlier information.
- The fields of dataToBeTimeStamped, messageDigestAlgorithm and additionalInformation equal to those in the request message.
- The field of requestedTime is the date and time when the server received the request. The server, actually the CVSTS authority, certifies that the time-stamped data had existed before this time.
- The field of generationTime specifies the date and time when the response is generated and signed.
- The field of responseIdentifier is the identifier of the response message.
- The field of serverCertificate is the PKC or certification path of the CVSTS server discussed in Clause 3.6.3 (2).

All the above information is digitally signed with the private key of the server,

and the signature can be verified with the public key contained in the PKC issued by the CVSTS authority.

3.5 Performance of CVSTS

In order to estimate the scalability of the CVSTS, the CVSTS server and client that handled the CVSTS request and response messages including a single PKC were implemented and the server performance was measured in the environment described in Table 3.1.

It takes 0.00575 seconds for the server to process a CVSTS request; 91% of the processing time is used for generation of the signature, and the remaining is communication processing, database processing, data decoding and encoding, etc.

For the purpose of the estimation of the scalability, the number of users of a messaging system that a single CVSTS server can serve is estimated according to the M/M/1 queuing model [CX1994].

Table 3.1: Evaluation Environment

Item	Name or Value
CPU	UltraSPARC-Ⅱ(296MHz)
Memory	512MB
OS	SunOS 5.6
Signature algorithm	Elliptic Curve Encryption
Key length	160 bits
Number of PKCs	131,072
Database	Ndbm contained in the OS

The mean service rate (μ) is 1 / 0.00575 = 174; in this case the processing time at the client side and the transmission time of the messages are ignored, because the target of the estimation is the scalability of the server. The mean arrival rate (λ) is reckoned from statistics of received messages of the organization with which the author is; a message gateway serving 3,705 people processed 151,981 in a week, so λ = 151981 / (60 × 60 × 8 × 5) = 1.06. The total waiting time is calculated with the equation $T = 1 / (\mu - \lambda)$ and the result is 0.00578 seconds.

The mean arrival rate is considered to be proportional to the number of users,

and it is possible to calculate the number of users in the case of 1 second total waiting time; the total waiting time is 1 second when the mean arrival rate is 173, and the corresponding number of users is calculated as 605,000.

Credit card authorization is a major business transaction, and it is said that a major credit company in the world dealt 16 billions transactions from July 1997 to June 1998. In this case the mean arrival rate is calculated as $\lambda = 16000000000 / (60 \times 60 \times 24 \times 365) = 510$. This value is larger than the mean service rate, and the server cannot handle such huge number of transactions.

3.6 Discussions

By synchronizing the validity period of the UAV with the personnel changes, PKCs are not revoked even if the owners move their departments. This reduces the size of CRL issued from an organizational CA. With the less revoked PKCs, the PKC information server of the organizational CA can send the update information to the master CVSTS server more frequently, and the service gives the fresher revocation information to users. The service also gives an evidence of the existence of a message, and this prevents repudiation of the message creation or sending by the originator. As a result, the UAV and CVSTS solve the problems described in Section 3.2. In the following other matters concerning the UAV and CVSTS are discussed.

3.6.1 Scalability of CVSTS

Evaluated in the previous section, a single CVSTS server can serve hundreds of thousands users of messaging system, and this performance seems to be poor taking account of the size of the Internet. The credit card authorization transaction exceeds the capacity of the server. With the adoption of the clustering technology to the slave CVSTS server, it can serve more clients and transactions. However, taking account of the network load balance and the reliability of the service, the introduction of multiple slave CVSTS servers is inevitable.

When multiple CVSTS servers exist, another problem arises; how to disperse the service requests among the servers? A technique exists to disperse accesses to the servers; when a BIND server [VP1996] is asked an IP address of an application server with its domain name, it returns one of IP addresses of the duplicated servers. The server does not take account of the metric between the servers and user clients in network nor the load of the servers, but it is a practical solution. In order to apply the

technique, the domain name of the CVSTS servers is embedded in the PKC with a newly introduced extension field, say CVSTSLocation extension field, and the CVSTS client accesses the server whose address is obtained from the BIND server. More sophisticated method is the single IP address architecture [SN1998]; the duplicated servers are assigned a single IP address, and a client accesses the nearest server in the sense of routing.

3.6.2 Synchronization of CVSTS Data

From the nature of the Internet, there is no guarantee of the Quality of Service (QoS), especially communication delay, and this makes the synchronization of the CVSTS local PKC information databases difficult. Possible solutions to the problem are as follows.

(1) Resource Reservation

The first solution requires change of the Internet architecture as well as routers and reservation establishment protocol. The technique of resource reservation [BR1997] can be applied to guarantee the distributed database some level synchronization. All the routers in the path between the PKC information server and the master CVSTS server and between the master server and slave servers must reserve resources for the transmission of the update information within a specific delay. With the combination of the differentiated services [BS1998], the resource reservation will be realized in the Internet architecture.

(2) Best Effort

The first solution requires change of routers in the Internet and will take some time to be realized. The second solution is ad hoc; the user judges the freshness of the PKC status with the lastUpdate fields in the response. If a user wants later status of a PKC, then the user accesses another slave server with higher service quality. In this case the quality is specified with the CVSTSLocation field.

ted with the signature of the server. As a result the information is safe as long as the private keys are safe.

(2) Measures against Key Compromise

The private keys of the CVSTS servers must be protected by using every considerable means, because the compromise of the keys leads to the collapse of the service. One of the methods to reduce the impact is to use different keys depending on the hash value of the response content, that is, the data to be signed by the server. The CVSTS authority needs to announce the rule of selection of the keys. The client needs to select the public key according to the selection rule to verify the response. With the rule even a private key is stolen, the attacker that stole the key can only forge CVSTS responses that match the rule of the stolen key.

Another measure to reduce the impact of the key compromise is to change key frequently. The shorter period of the private key usage leads to the less impact in the case of the compromise of the key. The key usage period is specified in the privateKeyUsagePeriod field [ISO1995]. In this case the client verifies that the generationTime of the response is within the period.

Both of the above two measures lead to the increase of the public keys of the CVSTS servers, and the client is required to hold the public keys to verify the response. In order to avoid such cost increase, the CVSTS authority establishes a CA issuing PKCs of the CVSTS servers, and the CVSTS server puts its PKC in the serverCertificate field of the responses. The client holding the public key of the CA does not need to hold the public keys of the CVSTS servers, and can verify the responses.

3.6.4 Cost of UAV

The use of the UAV helps to reduce the size of the revocation information, but it increases the cost of issue of the PKCs. Issuing procedure varies among CAs, but in the case of updating PKC issued from an organizational CA, the following steps can be adopted:

(1) A user creates a new key pair and update request of the new public key, and the user signs the update request with the old private key,
(2) the user sends the request to the PKC issuing server of the organizational CA,
(3) the server verifies the request, and
(4) the server creates a new PKC including the new public key and a new UAV retrieved from personnel database, and sends it to the user.

Comparing with the processing of the CVSTS server described in Section 5, the issuing server needs to verify the request, retrieve the new UAV additionally. The processing time can be estimated less than 0.02 seconds plus the retrieving time of the UAV. As a result if the retrieving time is not too long, the procedure of updating PKC described above is practical.

3.6.5 PKC for Encryption

Even if the key pair for signature changes frequently, there is no problem for the recipient as far as the PKC is available. But the situation is different in the case of the key pair for encryption, because the owner needs to hold old private keys to decrypt messages encrypted with the old public keys. Therefore it is desirable to use the same key pair for encryption as long as the key pair is safe.

As a solution for this problem, the key pairs for signature and encryption are separated, and the public key for encryption is distributed in another PKC. The two PKCs for signature and encryption can be distinguished with the keyUsage field [ISO1995]. Moreover the PKC for signature may include the pointer to that of encryption key in an extension field, for the sake of convenience to retrieve the PKC for encryption through a directory service.

3.7 Related Works

3.7.1 Attribute Certificate

The Public-Key Infrastructure (PKIX) working group of the Internet Engineering Task Force (IETF) discusses the Attribute Certificate (AC) [FS1999], which binds a public key owner and its attributes with the same technique of the PKC. The AC is mainly used to make access control decision or authorization, when an application client requests an access to resources on a server. The AC is proposed for the two reasons:

(1) Attributes of the public key owner do not have the same lifetime as the binding of the public key and its owner. If the attributes are put into the PKC, the lifetime of the PKC becomes short.
(2) The PKC issuer is not usually authoritative for the authorization information.

The UAV field is another solution for the validity period mismatch problem pointed out in the first reason. The signer attributes such as department name, title or address are required for the target applications, such as secure messaging, signed document, and business transaction. These attributes can be considered to be authorized from the same authority of the PKC issuer such as the organizational CA operated by its personnel and/or information system management department. Comparing the AC, the advantages of the UAV are the smaller cost to verify the AC and less impact of changing the current PKI.

3.7.2 Electronic Signature Timestamp Server

The idea of the TSS is not new; for example, [HS1991] discusses how to realize privacy of the signed document and how to prevent back-dating of time stamp record. From the view point of PKC status verification, the most related work is ESTS [LJ1995]; the service supports certification path verification, time stamp, data archiving and data retrieving. The differences between the ESTS and the proposed system are the following:

(1) Architecture of Overall System
The ESTS focuses on protocol between a user client and a server, but it was not studied how all the system should be composed in the large network environment like the Internet.
(2) Support of Verification Time
The ESTS verifies the status of the requested certification path at the time of request, on the other hand the CVSTS verifies the path at any time user requests. This feature is useful when the user wants to know the status at a certain time point, such as the time when the signature of a message was generated, because the recipient concerns that the PKC was suspended temporally at the time, or revoked after the signature was generated but before the recipient received the message.

3.7.3 Online Certificate Status Protocol

The PKIX working group is also discussing the Online Certificate Status Protocol (OCSP) [MM1999b] giving the PKC status. The CVSTS differs from the OCSP not only at the above two points but also in the following points:

(1) Time Stamp Service

The CVSTS supports additionally the time stamp service and consequently the non-repudiation service, and this is the most significant difference with the OCSP. As a result the request and response have fields that the OCSP does not include, such as dataToBeTimeStamped, requestedTime.

(2) Fields in Request

The requesterName, reqCert and nonce fields of the OCSP request correspond to the requestOriginator, certificates and requestIdentifier fields of the CVSTS request. The CVSTS does not have a field corresponding to the acceptableResponses of the OCSP, because the CVSTS supports a single response type. As for serviceLocator field, the CVSTS request does not have such field, because the CVSTS authority has all the revocation information of the hierarchical CAs described in Section 3.4.1, and does not need to route a request to another server.

(3) Fields in Response

The fields of producedAt, certStatus, thisUpdate and certs of the OCSP response correspond to the generationTime, certificates, verificationResult lastUpdate and serverCertificate fields of the CVSTS response.

The CVSTS does not have fields corresponding to the responseStatus and responseType fields of the OCSP. This is because the CVSTS assumes no errors such as request syntax or internal errors, and the CVSTS supports a single response type.

The client can confirm the name of the CVSTS server with the verification of the response, so that the name of server field such as responderID field of the OCSP is omitted.

The nextUpdate of the OCSP is used to indicate the next update time of the PKC information and the client does not trust the OCSP response when the time is past, because the indicated update information is not reflected to the response. In such case the CVSTS server returns the unknown result, instead of indication by the nextUpdate field.

The archiveCutoff field of the OCSP is used to indicate the cutoff date of retained

revocation information beyond the PKC expiration time. The CVSTS does not assume such archive and does not include the field.

3.8 Conclusions

In this chapter three problems of PKC revocation in the PKI, the unavailability of the latest revocation information, the large CRL size, and the lack of non-repudiation mechanism are described, and then two solutions are proposed; the UAV is suitable for an organizational CA in order to eliminate revocation caused by periodical personnel changes, and the CVSTS is an on-line service of combination of PKC verification and time stamp providing latest PKC status and non-repudiation services. A prototype of the service is implemented, the security and scalability of the service are discussed, and it is shown the service can serve 605,000 message system users within one second response time.

Internet Engineering Task Force (IETF) defines another time stamp protocol [AC2001b], which is basically same as the time stamp function of CVSTS. The protocol contains additional information for time stamp service; the accuracy field in the response specifies the time deviation of the time stamp. The CVSTS assumes the deviation is specified in the service level agreement of the service. IETF also defines Data Validation and Certification Server (DVCS) protocols which validates signature of signed data as well as the PKC status. The service encompasses the function of CVSTS. CVSTS is a basic service which contains the two services: a time stamp service which should be provided by a trusted third party, and PKC status providing service which solves the problems described the subsections 3.2.1 and 3.2.2. The CVSTS does not give validation service of signed data, because the user can validate the signature with the PKCs and status information in the CVSTS response.

Chapter 4

Securing Parts of Document

4.1 Introduction

While the previous two chapters treat problems and solutions of the management of private and public keys of the public key technology, this chapter describes encryption and signing of office document with the public key technology.

Office document is structured; it contains multiple chapters, clauses, pages, tables, figures, etc. The Open Document Architecture (ODA) [ISO1988] standard was developed to interchange documents in an open systems environment. Deliberately the standard does not consider the form of document transmission, via a network or with a floppy disk; it only specifies the form of an encoded octet stream called the Office Document Interchange Format (ODIF). In order to secure a whole ODIF stream, the general cryptographic envelops [RSA1993b, LJ1992] can be used, but such envelops cannot protect parts of document. On the other hand, the ODA security addendum [ISO1990] supporting partial encryption and signing is under standardization, but it is defined separately with the PKI standard [CCITT1988], compatibility with existing document editor.

In this chapter, working of the author during stay at University College London (UCL), the activities of which in ODA and PKI is given in Appendix F, is described; the consistency of the ODA security with the PKI standard and connectivity of implementation to existing document editor are evaluated. There are differences of syntax of the same data type and contradictions in the ODA security addendum. There is also a problem in integrating of an existing ODA editor with a filter program implementing the ODA security. Details of the differences, contradictions, and problem are described and resolutions are proposed after overview of the ODA.

The author is not concerned here whether the ODA as a standard eventually wins the day, but he believes that many of the considerations which he has addressed in implementing secured the ODA described in this chapter would require resolution in any attempt to secure parts of complex documents.

In this chapter, the definition of security attributes are changed; authenticity is separated into two attributes, integrity and authenticity, and non-repudiation is added

according to the definition of the ODA standard. Details are described in Clause 4.3.3.

Firstly, requirements for secure document interchange are described in Section 4.2, outline of ODA security in Section 4.3. Secondly, problems and resolutions in compatibility of ODA security and PKI, contradictions of the ODA security, and a problem integration with ODA editor are discussed in section 4.4. Next, application of the ODA security to document stores in Section 4.5. Finally, some conclusions are presented in Section 4.6.

4.2 Requirements for Secure Document Interchange

Document interchange can be divided into three categories: interchange between two parties, circulation within a small group, and dissemination to a large number of people. In the following paragraphs, the security requirements of each category are described.

4.2.1 Interchange between Two Parties

Secured document interchange between two parties is the normal message exchange (e-mail) with security additions; the following security services are required:

- Maintenance of the confidentiality of document content
- Assurance of the integrity of document content
- Assurance of the authenticity of origin of the document
- Maintenance of the confidentiality of document flow
- Provision of proof of the exchange (submission and delivery)

The last two services must be provided by the transfer system, such as the message or file transfer systems, and are out of scope of the document interchange security. The message transfer system may provide the first three services. Because the document security is required also after transmission, for example storing a document with a digital signature, it is desirable to support these services at the document level. Services securing a whole document may be adequate, because securing parts of a document, such as paragraphs or figures, makes no sense when the document is sent to only one recipient from the originator.

4.2.2 Interchange in Group (Group Cooperative Work)

In the case of interchange in a small group, security on parts of a document may be required. For example, as shown in Figure 4.1, the digital signature of a part of the document may be required because some parts may be written by multiple authors and other parts by another. Encipherment of parts of the document may be required because an author wants to restrict access to the parts to some members of the group. For these reasons at least confidentiality, integrity and authenticity of parts of a document are required. In addition, the following facilities may be required for group cooperative authoring:

- addition/modification/deletion of parts to/of/from an existing document while keeping signatures of other parts valid;
- addition of a signature to the part which has been already signed by other members;
- signing not only the content of parts of a document but also some other information such as time, location, signer's role, etc.

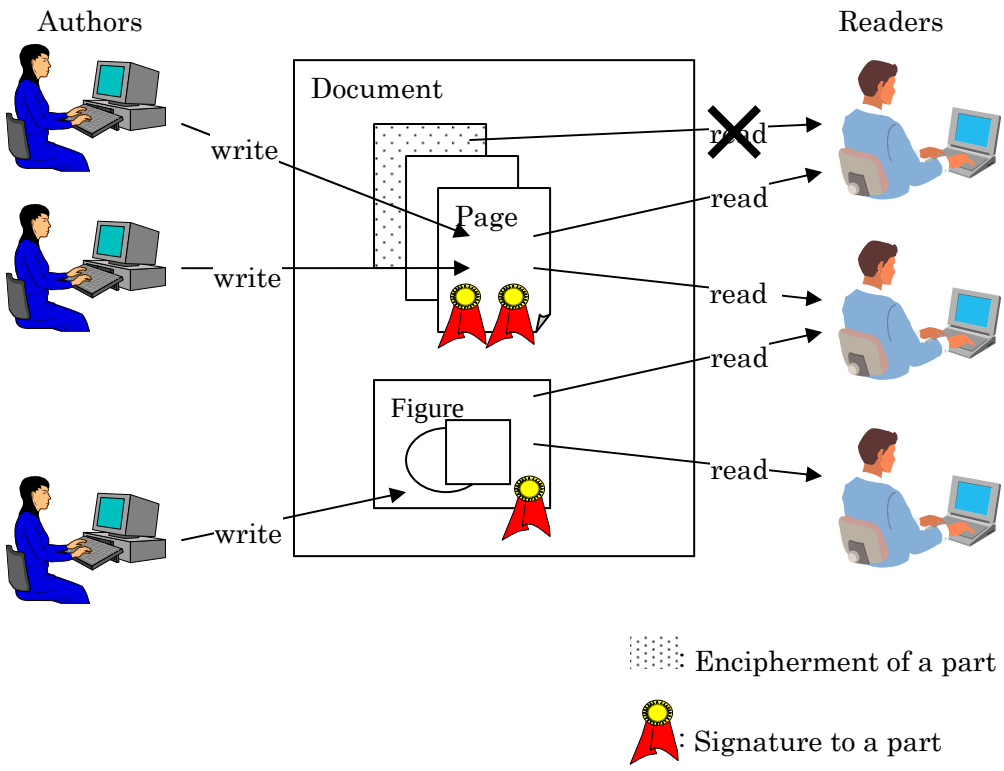


Figure 4.1 Security Requirements in Group

4.2.3 Interchange in Large Number of People

There are two typical applications in this category: one is a network news service or a public mailing list; the other is a document archive, search and retrieval service. In the former case authenticity and integrity may be required without confidentiality because the documents are public; an example is documents indicating how to combat virus issued by the CERT or a software supplier. In the latter case some access control mechanisms on the document archive server may be required; an example of access control is one based on user's identity, group, organisation which the user belongs to, user's clearance, etc. Securing a whole document is normally adequate during transmission from the server to the user. Signing parts of a document may be required when the document is written by multiple authors.

4.2.4 Requirements for Secure Document UA

As the first step of securing document interchange, it was decided to start with the implementation of a document UA. This is because there is a standard which specifies the secured ODIF and there has been already a non-secure document UA, while not having formulated detailed requirements for a document archive server. The next steps are to specify the requirements for a server, design access control mechanisms and the implement. Security services and facilities for a secure document UA are follows:

- confidentiality, integrity, authenticity and non-repudiation of origin of a whole, or parts of, a document;
- addition, modification and deletion of parts of a document while keeping signatures of other parts valid;
- addition of a signature to a part of a document;
- signing the content with other information

Security services of a whole document can be implemented with the DOCSEC [GS1990] or PKCS#7 messages [RSA1993b]. The DOCSEC supports the four security services of whole documents and arbitrary octet strings. PKCS#7 defines a cryptographic message syntax which supports the services as well as other syntax; the enciphered content with or without encryption keys, the content with signatures, etc. Both have been implemented as filters, using OSISEC for the application of security; there are two programs one of which encodes plain data into enciphered and digitally

signed data and the other decodes the secured data into the original plain data.

For security services on parts of a document, the ODA security addendum [ISO1990] is adopted. This addendum supports the above security services on parts of a document, that is, ones of the document profile and the bodyparts. Security functions must be implemented to support as many existing document UAs as possible. The security information which users concern should be displayed, or if necessary input in a user-friendly manner.

4.3 ODA and Security

4.3.1 ODA Concepts

The ODA standard [ISO1988] describes an abstract view of an office document and a document processing model as well as an interchange format of a document. A document consists of components, that is, the document profile, generic structures (logical and layout object classes), specific structures (logical and layout objects), styles (layout and presentation styles) and content portions. These components give two views of the document; a logical structure which represents a logical view of the document such that a letter header consists of a date, an addressee, a subject, etc., and a layout structure which represents a layout view of the document such that a letter header page consists of a logo frame, a date frame, etc.

The ODA standard defines three kinds of document form; a processable form with logical structures is created after an editing process, a formatted form with layout structures is produced by a formatted process, and formatted processable form with both logical and layout structures is also produced by a formatted process. An imaging process takes a formatted or formatted processable form and produces a final document. All these document forms are encoded into the ODIF stream in which all components are represented as sets of attributes.

4.3.2 Security of ODA Document

In the security addendum [ISO1990] to the ODA standard, two concepts of document security are provided. While securing the whole document which is out of scope of the addendum, it suggests that the security policy of the domain to which the originator belongs may specify how the whole document should be handled as a single unit. One method to realize this is to use a secure transport mechanism such as a secure X.400 or

a secure File Transfer and Access Method (FTAM), etc. The second concept is the securing of parts of the document. A part means any part of the document profile or of the document body. The addendum does not specify any particular algorithm or scheme, but it provides the means to protect parts of the document. In the rest of this section the outline of the ODA security features are described, and the new attributes added to the document structures are introduced.

4.3.3 Security Features

The security features supported in the addendum are confidentiality, integrity, authenticity and non-repudiation of origin; each is described below.

- Confidentiality of specified parts of a document is achieved by enciphering specified parts, so that only privileged recipients can decipher and read them.
- Integrity is demonstrated by ensuring that specified parts of a document cannot be changed or destroyed in an undetectable manner. The privileged recipient can verify that the specified parts have not been altered since the originator sealed them. The certainty provided by this property is limited to a detection of change; the replacement of the whole sealed parts and the seal itself cannot be recognised. Nevertheless, by sealing the whole document, tampering with its parts can be recognised.
- Authenticity is demonstrated to a privileged recipient, by allowing him/her to verify that the source of specified parts of a document is the claimed one.
- Non-Repudiation of Origin assures that a person cannot deny being the source of specified parts of a document. Non-repudiation of origin can also be achieved by adding seals to the parts.

4.3.4 Protected Part Structures

Four new kinds of protected part structures are added to document components. The first structure includes a part of the document profile to be sealed, and the other three structures include enciphered parts of the document profile, the document bodypart enciphered before and after a layout process.

- Sealed Document Profile Descriptor
- Enciphered Document Profile Descriptor

- Pre-enciphered Document Bodypart Descriptor
- Post-enciphered Document Bodypart Descriptor

4.3.5 Attributes of Document Profile

A new attribute, document security, is added to the document profile. The sub-parameters of this attribute are described in the following.

- Oda Security Label
  This attribute specifies the ODA security label of the document, that is, security policy, sensitivity, protection level, etc.
- Sealed Document Profiles
  This attribute includes information about integrity, authenticity and non-repudiation of origin of parts of the document profile.
- Sealed Document Bodyparts
  The protected information about sealing of bodyparts before a layout process can be found in the pre-sealed document bodyparts attribute. When document bodyparts are sealed after a layout process, the post-sealed document bodyparts attribute is used.
- Enciphered Document Profiles
  This attribute specifies information concerning each enciphered part of the document profile.
- Enciphered Document Bodyparts
  The attribute pre-enciphered document bodyparts specifies the information concerning each document bodypart enciphered before a layout process. When a document bodypart is enciphered after a layout process, the post-enciphered document bodyparts attribute is used.

4.3.6 Attributes of Document Structure

Two new attributes are added to the document structures and styles.

- Sealed
  The sealed attribute is added to generic and specific structures and styles. This attribute specifies sealing status and identification of the sealing information of the sealed bodypart.

- Enciphered

  The enciphered attribute is added to generic and specific structures. This attribute specifies enciphering status and identification of the information about the enciphered bodypart.

4.4 Problems and Solutions

In general, OSI standards are defined without prior implementations; as a result ambiguities and contradictions with other standards are often introduced. This section outlines certain such problems uncovered during the integration of PDOCSEC and the Slate editor in the PASSWORD project at UCL which is described in Appendix F.

4.4.1 Sequence of Constituents

In the addendum, the order of constituents of sealed bodyparts is specified but how they should be encoded is not. The following ASN.1 type is used to encode constituents of the bodyparts. An agreement is necessary to establish interoperability between different systems.

```
Sequence-Of-Bodyparts ::= SEQUENCE OF CHOICE {
    layout-object-class         [1] IMPLICIT   Layout-Class-Descriptor,
    layout-object               [2] IMPLICIT   Layout-Object-Descriptor,
    content-portion             [3] IMPLICIT   Text-Unit,
    logical-object-class        [5] IMPLICIT   Logical-Class-Descriptor,
    logical-object              [6] IMPLICIT   Logical-Object-Descriptor,
    presentation-style          [7] IMPLICIT   Presentation-Style-Descriptor,
    layout-style                [8] IMPLICIT   Layout-Style-Descriptor
}
```

4.4.2 Sealed Attributes

When a part of a document body is sealed, the sealed attribute of the constituent must represent the sealing status. The addendum specifies the attributes indicate “current status” of sealing, but does not specify whether these values of the attributes are included or not in the message digest. Because the message digest result depends on the values, an agreement is necessary to establish interoperability between different

systems. The current PDOCSEC calculates the message digest after the values are set.

4.4.3 Personal Name

In the current PDOCSEC system, asymmetric encryption is used to seal the content and to encrypt a content encryption key which is used to encrypt the content, and a public key is retrieved from the X.509 certificate issued from a CA. The X.509 recommendation [CCITT1988] defines Distinguished Name (DN) to identify a user. Example of DN is as follows [KS1995]:

CN=Yoshiki Sameshima, OU=Research & Development, O=Hitachi Software, C=JP

which represents an entity which name is "Yoshiki Sameshima" belonging to an organizational unit naming "Research & Development" of an organization "Hitachi Software" in Japan. However the ODA standard adopts Personal Name as a name of a user which includes a surname, a given name, initials, and a title. A user in a small community such as an organization could be identified with the information and a certificate or a trusted public key can be retrieval from a local cache or a directory, but this is impractical in a large scale community.

To solve this problem a new parameter name with type Distinguished Name is added to the Personal Name type.

```
Personal-Name ::= [APPLICATION 6] IMPLICIT SET {
    surname                 [0] IMPLICIT   Character-Data OPTIONAL,
    givenname               [1] IMPLICIT   Character-Data OPTIONAL,
    initials                [2] IMPLICIT   Character-Data OPTIONAL,
    title                   [3] IMPLICIT   Character-Data OPTIONAL,
    name                    [4] IMPLICIT   DistinguishedName OPTIONAL
}
```

By using this parameter a privileged recipient can get an originator's certificate via the directory service to verify its seal and an originator can get privileged recipient's public key for encipherment. However, there is still a problem because the directory service is not yet sufficiently ubiquitous that every recipient can get certificates from it.

4.4.4 Certification Path

One solution to getting an originator's certificate without directory access is to send the certificate path explained in Section 4.4 with the sealed information. Some applications use this method; for example, a PEM message may include Originator Certificate and Issuer Certificate fields, and an X.400 message has a certificates attribute in the envelope for this purpose. However there is no attribute which corresponds to the above attributes in the addendum.

To include a complete forward certification path, the Certification Path attribute imported from the X.509 standard is stored in a sub parameter of the sealed document profiles and sealed document bodyparts attributes. If the privileged recipient has the top level CA's public key, then he/she can get the originator's public key and verify the seal. This is useful when documents are stored for a long term, because it is necessary to store only the top level CA's public key and not necessary to store other information such as users' public keys. There still remains one problem. If the security policy of the privileged recipient's domain enforces a check of the originator's certification revocation list [ITU2001], the privileged recipient must access the list by some method.

4.4.5 Seal Data

The Sealed Document Profiles and the Sealed Document Bodyparts attributes include plain message digests of parts of the document. However when a part is enciphered, the message digest of the part should be encrypted to prevent an adversary from deducing the content. This is because the adversary can determine which of a list of candidate content (e.g., "yes" or "no") is the actual content by comparing message digests of them to one in the attribute. This is also true for signature, if the signature algorithm is giving message recovery. Other information, time and location, may be required to be encrypted; an agreement is necessary to establish interoperability.

The current implementation encrypts the message digest and signature, actually the 'seal-info' and 'seal' field of the 'Seal-Data' data structure illustrated in Figure 4.2, using the DES-CBC algorithm [NIST1977], and the key information encrypted with the privileged recipient's public key is stored in a sub-parameter of the attribute. Only the privileged recipient knows the actual message digest value.

4.4.6 Method Information

The Method Information type, which specifies the algorithm used for generation of a message digest, an enciphered document part, a seal, consists of two parameters; the object identifier and the character string which identifies the algorithm. However this definition is inappropriate when the algorithm requires parameters. In addition this type is incompatible with the Algorithm Identifier type defined in the X.509 standard which consists of two parameters, the object identifier identifying the algorithm and the required parameter to the algorithm.

4.4.7 DES-CBC Key and IV

The current implementation uses the DES-CBC algorithm [NIST1977] with a randomly generated pair of a session key and an initialization vector for the encipherment of parts of a document. Before the encipherment of the key pair, it must be encoded in accordance with some method. A typical method is that only the session key is enciphered, while the initialization vector is handled as the parameter of the algorithm. However the current PDOCSEC stores the pair in the following ASN.1 type, BER-encodes and enciphers it, because ODA's Method Information type does not have the parameter field. As a different system may encode in other way, an agreement is necessary to establish interoperability.

```
DES-CBC-Information ::= SEQUENCE {
    deskey                  [0]  IMPLICIT  OCTET STRING,
    vector                  [1]  IMPLICIT  OCTET STRING
}
```

4.4.8 Protection of Document Profile

When a part of a document profile is sealed, the part is stored in a sealed document profile descriptor. When a user enciphers and seals the same part of the document profile, the encipherment is ineffective because the ODIF stream contains the plain part of the document profile in plain form illustrated in Figure 4.2. The 'Sealed-Doc-Profile' data structure, actually 'Sealed-Doc-Prof-Descriptor' data structure, contains the plain document profile attributes which are the objects of signing. These attributes are also included in encrypted form in the 'Protected-Doc-Parts' data structure.

To avoid this problem the plain sealed document profile descriptor is not included in the secured ODIF stream; the 'Sealed-Doc-Profile' data structure contains the encrypted data which is also contained in the 'Protected-Doc-Part' data structure as illustrated in Figure 4.3. As described in The verification process deciphers the corresponding enciphered part of the document profile and calculates the message digest. With this solution it is possible to seal and encipher the same part of the document profile.

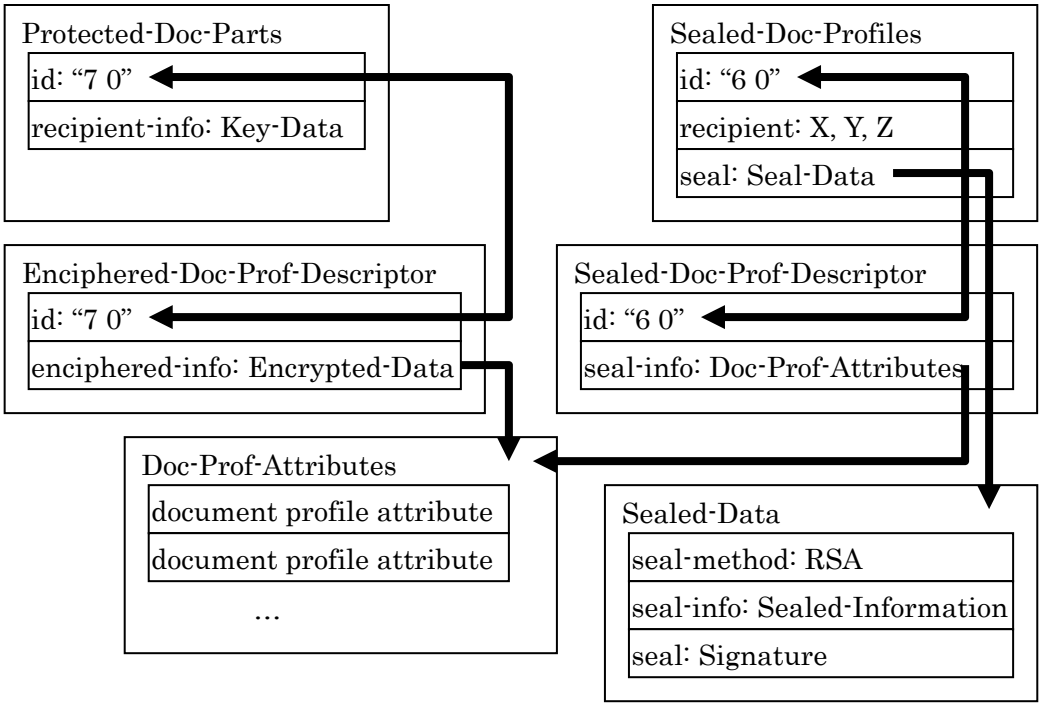


Figure 4.2: Encryption and Signing of Part of Document Profile in Addendum

4.4.9 Sealed Information

A seal is calculated from a message digest of the content, the sealed time, the location and the originator's personal name and other information cannot be included into the seal. For example the following information may be required according to a security policy:

- authenticated time
  This is time information authenticated by a third-party or a timestamp server. The time in the addendum is claimed by the originator.

- role name

  The Personal Name includes only title information as far as information of a person's attribute. However it is desirable to include the role, the department name to which the originator belongs to, etc.

To include any information in the seal, it is better to change the type to be sealed to a sequence of attributes such as the authenticated attribute type of PKCS#7 [RSA1993b].

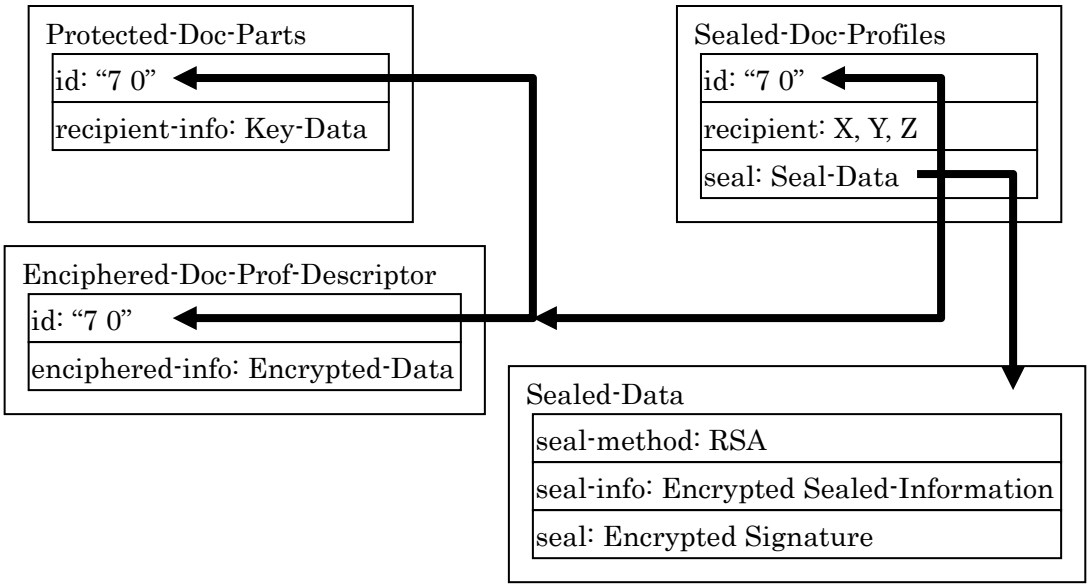


Figure 4.3: Proposed Encryption and Signing of Part of Document Profile

4.4.10 Integration with Slate UA

Two of the functions described in Section 4.2.4 are required. However addition and deletion of parts while keeping signatures of other parts valid and signing to additional information is not implemented. In the following the reasons and solutions are described.

(1) Addition and Deletion of Parts

Addition of a new part and deletion of a part are not implemented because OCIDs are changed from originals and this makes seals invalid. For example, assume that the object identifier of the first paragraph is "3 0 1" and the second one's is "3 0 2". If a new

paragraph is added between the two paragraphs the Slate-ODA filter assigns “3 0 2” to the new paragraph and “3 0 3” to the old second paragraph and seals containing objects which OCID is greater than “3 0 3” are not valid. It is same in the case of a deletion of a part from a signed document. This is because the Slate editor does not know OCIDs at all, and the Slate/ODA converter assigns new OCIDs to the sealed parts. However this is not a particular problem to the Slate editor and Slate-ODA converter. This is a common problem of existing editors which use converters convertors to transform ODIF to the editor original formats and vice versa. There are several possible solutions:

- Generate the message digest of a bodypart without OCIDs.
- Assign OCIDs in advance for later addition of parts.
- Keep mapping information from newly assigned OCIDs to old OCIDs (the identifiers when the original message digest was calculated) in a parameter and use the old OCIDs during the verification time.

(2) Signing to Additional Information

Signing time and location can be sealed because the addendum includes these attributes in the Sealed Information but other information such as authenticated time is not supported because the Sealed Information is not extensible as mentioned in the previous section.

4.5 Application to Document Stores

This chapter does not describe in detail how these techniques are applied to document archive servers. The security during transmission can be implemented with PDOCSEC, DOCSEC, or PKCS#7. The difficulty is the authorization mechanism when there are large numbers of users and documents. In UCL, an authorization mechanism based on capability scheme which uses PAC (Privilege Attribute Certificate) is under implementation and it will be applied to a document server. The project implements access facilities to this secure ODA documents database. The access methods use the Wide Area Information Service technology [KB1989] (a text retrieval system based on ANSI Z39.50 [ANSI1988]) - but with certain modifications to deal with access control. The system has been integrated with the BBN Slate document editor. The database’s access environment is able to enforce personal access rights for read and search operations founded on authentication of the user identity. The service enables users to

trust information that they extract by allowing database user agents to check the integrity of the information supplied. This is done by authenticating trusted source servers. The basic WAIS client functionality has been improved to work with the WAIS server which manages the secure database. The Secure WAIS Database Client is a powerful client, which offers users access to the full range of Secure WAIS Database Server services. From a security point of view, the Client/Server environment actually covers the following requirements:

- ensuring the remote user that the received document has not been tampered with (document content integrity),
- proving that the server is the source of the received document (document authenticity and non-repudiation of origin), (this property is only applied on demand, in order to offer reasonable look-up speed):
- ensuring that the given document is only disclosed to the specified remote users (document content confidentiality).

Authentication credentials for users are based on the RSA public-key Cryptosystems [RSA1993b], where trust semantics are established through the use of key certifications defined in the X.509 Security Framework [CCITT1988]. Local mechanisms are required to maintain the confidentiality of user's private key.

4.6 Conclusions

In this chapter, compatibility problems between the OSI security and the PKI, contradictions in the OSI security, and problems in integrating the OSI security with an existing document editor have been pointed out. Resolutions for the problems and an application to storage have also been shown. Further work is needed, however, to ensure that the methods described do not incur unacceptable overheads in performance, and that the whole technology is sufficiently convenient and rugged for practical application.

Currently there are two standards of structured office document format, Open Document Format for Office Applications (OpenDocument) [OASIS2007] of OASIS standard and Open Office XML [ECMA2006] of ECMA standard. OpenDocument supports encryption of whole documents, but does not support encryption nor digital signature of part of document. Open Office supports digital signature of part of document, but does not support encryption of it. These two will be major office

document formats, because some major word processor programs, such as OpenOffice.org and Microsoft Office, support one of the two formats, and a convertor of the two formats is developed as an open source program.

Chapter 5

Authorization with Security Attributes and Privilege Delegation in Multiple Domains

5.1 Introduction

This chapter focuses on access control or authorization in distributed environment, which has been paid less attention than other security elements such as encryption and digital signature of data.

In the recent few years, a number of information systems such as World Wide Web (WWW) [BT1993], Gopher [AF1993], and Wide Area Information Server (WAIS) [MP1992], have deployed widely and various kinds of information are stored and retrieved over the Internet. Security of network data, for example authentication, integrity check, and confidentiality of the Hypertext Transfer Protocol (HTTP) traffic, is realized with the Secure Sockets Layer (SSL) protocol [HK1995].

However, there is less deployment on access control or authorization of access; authorization is a process to grant an access of a subject, such as a human user or a client entity, to an object, such as a file or a server entity. For example, an OS realizes authorization of file access based on the ACL; an ACL entry, which is attached to an object, consists of a subject identity and permitted types of operation to the object, and reference monitor [DD1983] within the OS decides whether a subject is allowed or rejected to access the object by referring to the ACL. However, the application of the ACL scheme to the information servers does not seem to work, because the network to which such server is connected is divided to multiple security domains; a security domain is a collection of users, computers, and other resources, which are under management of a single authority. The reference monitor can authorize access of user to resources which belong to the same domain, but it cannot make decision if one of a user or resource does not belong to the domain, because the reference monitor does not have knowledge to judge.

In order to authorize access of subject to object which do not belongs to the same security domain, the Privilege Attribute Certificate (PAC) [YH1995, KP1994] or proxy (certificate) [NC1993] is used, which conveys privilege of the subject to a server

across the boundary of security domains, and the reference monitor of the server judges the access to be allowed or disallowed.

Delegation of privilege is an important element of security of a distributed system; the delegation happens when an entity asks another entity to work for the first entity as illustrated in Figure 5.1. In Domain A, User X accesses Server F, then a process which has the user's privilege starts after the authentication of user X (1). When the process accesses local resource, the access is permitted after authorization of the Reference Monitor (RM) of the server (2). When the user asks a job which requires resource in Domain C to Server G in Domain B (3), the process in Server F sends a PAC containing privileges of User X with the job request. Then the service process of Server G access the resource in Domain C with the PAC (4).

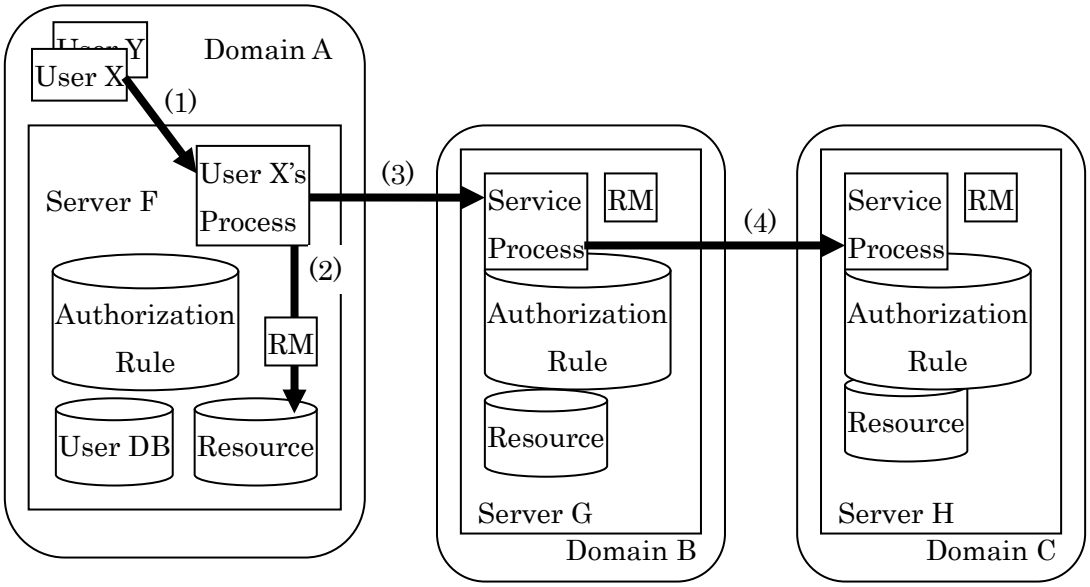


Figure 5.1: Privilege Delegation crossing Domain Boundary

Current delegation mechanism adopts chained PACs or proxies which contain privileges and restrictions on the privileges and are sent to a target, service process in Domain C in Figure 5.1, [KP1994, GM1990, NC1993]. The service must verify the chained PACs or proxies and decides whether an accessing entity, has access permission or not.

However, the authorization and delegation mechanism has several problems. Firstly, the restriction on privilege contained in the PAC is not formulated; in order to prevent the delegated server from abuse of the privilege, the PAC contains restriction on

privilege, but the method of interpretation of the restriction is not formulated, and this leads to the problem of interoperability of different security domains. Next the verification of the chained PACs may fail, because the protection mechanisms of PACs, the privileges and the restrictions may be different in each domain. Thirdly authorization rule is not formulated; the interpretation of privilege of the reference monitors in a domain should be unified, and it is desirable to represent the authorization rule in a formalized form. The situation is illustrated in Figure 5.1.

In this chapter, the meaning and interpretation of privilege and restriction in the PAC are formulated to solve the first and third problems. A new delegation mechanism is also proposed to solve the second problem, which uses a directory service to retrieve security information of other domains and translates the chained PACs to a single one which the server can verify and check the permission effectively.

Section 5.2 illustrates access control requirements for information servers in an organizational network and in a large scale network, and points out their features. Section 5.3 describes deficiencies of the ACL scheme and proposes the PAC/CAP scheme which is proper for the requirements. After problems of the current delegation mechanisms as well as a solution are given, an implementation of an authorization function and a document server which uses the function are described in Sections 5.4 and 5.5.

5.2 Access Control Requirements for Information Servers

The following two sections illustrate authorization requirements for such information servers.

5.2.1 Information Servers in Organizational Network

The followings are requirements for a server running on an organizational network which may consist of thousands hosts and provide information to thousands users in the organization, such as document servers, personnel information servers.

- The server restricts a user's access depending on a user's identity, a role, a group, an authentication level, a network location etc.
- The server controls a user's access according to categories of the information and

need-to-know of the user; for example, people in a development department may not be permitted to access accounting information because their need-to-know is different from the category of the information.

- The server grants a user's access by comparing the clearness of the information and the clearance of the user; for example, very few people can access "top secret" information.
- The server may permit a delegated access from an entity which works for a user who has the access right; for example, a printing entity may be enabled to read a file for which the requesting user has the read permission.
- The organizational network may be divided into a number of security domains, where the categories of information, the role names may be different.

5.2.2 Information Servers on Large Network

The followings are requirements for a (commercial) information server which runs on a large network and provides information to a large number of users. Examples of such a server are WWW, Gopher and WAIS servers in the Internet which store various information, such as on organizations, people, research or commercial activities, press releases, products, documents, video, etc.

- A user may pay a fee and get read permission to a set of information on the server.
- The server may distinguish users with their privilege classes which may be different depending on the fee; for example, a privilege class "A" user paying a large fee may be allowed to access a larger set of information than a privilege class "B" user.
- Only a small number of managers have write or modify permission.
- Each user's privilege has an expiration date.
- The number of information units may be very large but the units may be divided into less number of classes from the viewpoint of authorization management.
- A user whose payment is small may be restricted on the access ability or availability, for example available when the server's load is low.
- A user may access the server from other security domains, often with security policies different from those in the server's domain. As a result, the user's client and the server may support different sets of encryption algorithms and trust different authorities.

5.2.3 Features of Authorization

While in the ACL scheme the access is controlled with the identity of the user or the group to which the user belongs and the operation type, the reference monitor or the ACDF (Accesses Control Decision Function) [ISO1994], is required to authorize the access depending on various attributes of the user (subject), the information (object) and context such as an access time or a subject's location. As a consequence, it is necessary to represent varying characteristics of the subject and the object, and to decide the access control according to the characteristics.

5.3 PAC/CAP Access Control Scheme

5.3.1 Mismatch of PAC and ACL

The OSF/DCE security architecture [YH1995] and the SESAME architecture [PD1993] have adopted the PAC to specify and exchange the subject's characteristics; a unit of the characteristics is called a privilege and represented in the form of a security attribute, which consists of its type, value and additionally an authority which provides the semantics of the attribute. A set of privileges with control information is signed by an authority of the security domain to which the subject belongs [ISO1993]. The signed PAC is distributed to the subject and the subject presents the PAC to the application server. The server verifies the PAC and grants the subject's access depending on the privilege information and the access control information of the object. The PAC has the following properties:

- The PAC provides privileges of the subject, such as a role of the subject, groups to which the subject belongs, need-to-know of the subject.
- The PAC may specify conditions which the subject should satisfy, such as the minimal authentication level, network location (access point).
- The PAC may restrict characteristics of the object for which the PAC is valid, such as an object's name, a service type which the server supports.
- The PAC may specify a time interval in which the PAC is valid.
- The PAC contains not only privileges of the subject but also restrictions on the use of the privileges; restrictions may be added when the privileges are delegated from the original subject to another entity which works on the behalf of the subject. A typical example of the restriction is an access type such as "read only." The next

section discusses the delegation of the privileges.

- The PAC is signed by the authority to prevent unauthorized modification by the subject or during transmission and to make sure such PAC is valid.

While the privilege attributes can represent the characteristics of the subject, there is no appropriate representation of characteristics of the object. The OSF/DCE architecture has adopted the ACLs that specify which subject, group or role is permitted to access the object with types of operation [OSF1992]. However, authorization based on context information such as time, subject's location or a combination of such information cannot be represented with the ACLs. Moreover the ACL does not directly match the privilege because the privilege represents the subject's right of access, while the ACL represents permitted operations of the subject.

5.3.2 Combination of PAC and CAP

The Control Attribute Packages (CAPs) [ECMA1989] can solve the mismatch between the privilege attributes and the ACLs. A CAP, which is attached to an object, is a sequence of security attributes which represents characteristics of the objects, or characteristics of subjects which are permitted to access the object.

The following shows an example of CAPs which requires that the subject's need-to-know should include "accounting" and the subject's role be "manager," and its location should be in "local network" or strongly authenticated.

category = Accounting, Role=Manager, SubjectLocation=LocalNetwork;
category = Accounting, Role=Manager, AuthenticationLevel=Strong

While the role information can be handled as a subject identifier in the ACL, the category and the context information, such as subject location, authentication level, cannot be represented with the ACL.

The original document of the CAP specifies the data type of the CAP but does not specify how to compare the security attributes in the CAP and the PAC, nor how to grant access according to them. The authors have refined classes and semantics of the security attributes in the PAC and the CAP, and specified how to make the access control decision; the security attributes have been divided into the following six classes:

- Privilege

  A subject's privilege is represented with a privilege security attribute and transmitted in the PAC. The privilege represents access right or capability of the subject. Examples of the privileges are subject's clearance, need-to-know, role, group to which the subject belongs, etc.
- Positive Restriction

  A restriction is represented with a positive or negative restriction attribute and transmitted with the privileges in the PAC. A positive restriction attribute describes a restriction in positive form, that is, represents a restricted privilege. Examples of positive restrictions are permitted access time, subject's location, validity time of the PAC, subject's name, object to delegate, target name or type (ex. file name, server service type), etc.
- Negative Restriction

  A negative restriction attribute expresses disallowed privilege including. Examples are prohibited access type, time or access point, etc.
- Condition

  Security attributes in the CAP are divided into two classes: condition and exception. A condition security attribute of an object characterizes the object, a subject which access to the object is permitted, or status which must be satisfied. Examples are clearness, categories, required minimal authentication level of the subject, access time, etc.
- Exception

  An exception security attribute of an object is a negative form of the condition; it describes a characteristic of a subject which is prevented from accessing the object, or prohibited context status including. Examples are prohibited access time, disallowed access point (subject's location), etc.
- Context

  Context information is represented with context security attributes, which are maintained by the server or the ACDF, including authenticated subject name, authentication level, access count, seal algorithm of the PAC, authority name that issued the PAC, charging identity included in the PAC, access type, arguments of the operation, server's load, etc.

The ACDF is called with six arguments, namely privilege, positive restriction and negative restrictions in the PAC, condition and exception attached to the object being accessed by the subject, and context information. Each of the condition and

exception security attributes is compared against the attribute in the privilege or context class, and each of the restriction security attributes against one of the condition or context class, and the access is allowed when the all comparisons succeed.

An attribute may have an ordered value, which matches a higher or lower value of another attribute. A typical example of the ordered value attributes is the combination of the clearance privilege and the clearness condition; both attributes take one of values of "unmarked," "unclassified," "restricted," "confidential," "secret" or "top secret" (in ascending order), and a read access is permitted when the subject's clearance privilege is equal to or higher than the object's clearness condition. Another attribute has a time or a time interval value; for example, a value of the PAC validity time of the positive restriction class is a time interval and compared with the current time of the context class, and the comparison succeeds when the time interval contains the current time. The rule of the comparisons, that is, which attribute is compared with which attribute and its method is described in tables; description and its semantics are given in Clause 5.5.1. The combination of the PAC and the CAP has the following:

- Simple Semantics
  The semantics of the combination is simple and easy to understand; it is necessary only to compare attributes of the condition, exception, positive and negative restriction classes with corresponding attributes.
- Easy Management
  A security domain may require non-standard security attributes and the manager of the domain needs to configure authorization rules. With the PAC/CAP scheme, this is an easy task because the manager only needs to specify matching rules of the security attributes. A configuration example is illustrated in a later chapter.
- Checking Parameters of an Operation
  Grant of an operation may depend on the parameter of the operation as well as the operation type. For example, modification of salary to a value exceeding a specific amount may require an extra privilege. This authorization is realized by comparing a parameter attribute of the context class against a limit attribute attached to the object of the condition class.
- Various Syntax
  An attribute value is not limited to a string or an integer; it can be an arbitrary type such as a time interval.

On the other hand, the new combination has the following disadvantages:

- Efficiency

  The PAC/CAP scheme is more complicated and slower than the ACL scheme. However, each attribute of the restriction, condition and exception classes can be compared in parallel and it is possible to make the decision in a reasonable time.
- Mismatch with Underlying OS

  Many operating systems support the ACL for the authorization of file access; this might cause mismatch with the CAP.

5.4 Privilege Delegation across Domain Boundary

5.4.1 Current Mechanisms

It is a common requirement in a distributed system for a subject entity to request another entity to act for the subject on its behalf. The subject is called an initiator, the requested entity an intermediate, and the object a target. The most typical example is a printing service; an initiator asks a printing scheduler to print a file, the scheduler allocates the task to one of printer servers, and the assigned printer server reads the requested file from the target file server and prints it. In this case, the second intermediate, the printer server, needs the read permission of the file on the target.

The SESAME architecture [PD1993] has adopted the PAC chaining method; the chained PACs, which represent the delegated privileges, are included in the PAC for the intermediate. This makes the intermediate can use the initiator's privileges with any action and makes it possible to trace the delegation route which is required for auditing and charging. The initiator may want to make sure that the intermediate cannot use the privileges for other purposes than the requested action. This is accomplished by specifying restrictions on the privileges. Typical examples are restrictions of access type (read + write → read only) and target (non-restriction → specifying a target or a target service type).

In the Distributed System Security Architecture (DSSA) [GM1990], an initiator generates and signs a certificate to allow the intermediate to act on the initiator's behalf. Restrictions on time are included in the certificates, however, ones on targets or access rights are not well formalized.

A similar method, which is based on proxy, is proposed in [NC1993]. A proxy is a certificate that allows the intermediate which has the proxy key to operate with privileges of the initiator which granted the proxy. The proxy is protected from

unauthorized modification by adding a seal of the grantor. The following three proxies illustrates chained proxies, which implement a delegation from *initiator* to *intermediate*2 via *intermediate*1 where $[I]_{K\_X}$ stands for information I sealed with key of X:

original proxy: $[privilege, K\_initiator]_{K\_authority}$

delegated proxy1: $[restriction1, K\_intermediate1]_{K\_initiator}$

delegated proxy2: $[restriction2, K\_intermediate2]_{K\_intermediate1}$

The top proxy specifies that the authority of the security domain permits initiator's *privilege*, the next proxy sealed by *initiator* specifies that *initiator* allows *intermediate*1 to use privilege with *restriction*1, and finally the last proxy designates that *intermediate*1 grants *intermediate*2 to use the privilege with *restriction*2. All three proxies are sent to the target, which verifies the proxies and checks whether *intermediate*2 has or not privilege with restriction1 and *restriction*2.

5.4.2 Deficiencies of Current Mechanisms

The above delegation mechanisms have the following deficiencies in the case that the authority, the initiator, the intermediates and the target do not belong to a same security domain:

- Policy and Authority
  PACs are issued from a Privilege Attribute Server (PA-Server) in the SESAME architecture, and proxies are generated by the initiator, the intermediates or an authorization server in the proxy-based authorization scheme. However, the target domain does not accept the PACs or the proxies because the security policies of the domains may be different. Moreover the verification of seals generated by entities or authorities of other security domains may fail since the target may not trust authorities in other domains.
- Seal Algorithm
  The seal algorithm of the PAC may be different in each security domain and the

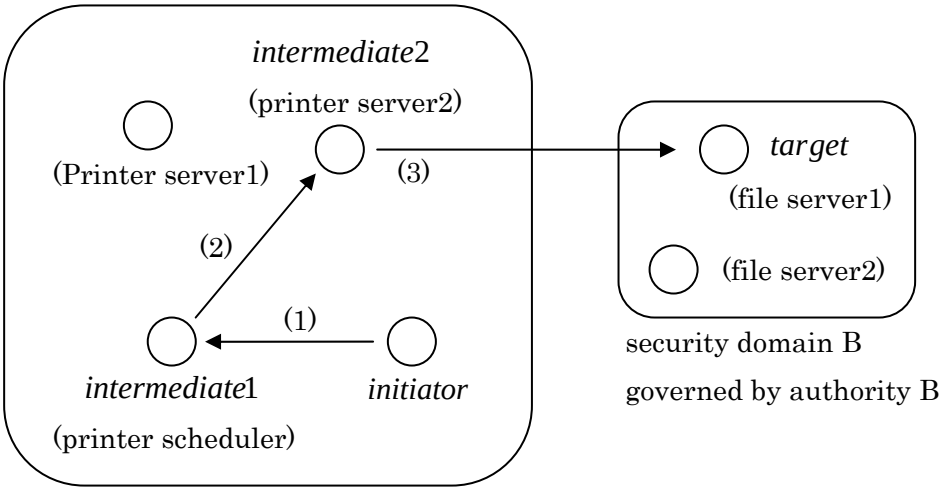
target may be unable to verify the seal and thus to check the chained PACs or proxies.

- Privilege and Restriction Mapping

  Each security domain may define privileges and restrictions of its own. Again the target may fail the translation of privileges or restrictions of the different security domain to local ones.

- Complicated Computation of Restrictions

  The checking of the chained privileges and restrictions may be complicated and cost much. For example, in Figure 5.2 the original privilege, privilege, does not have a restriction on the target, restriction1 added by initiator limits the target to printing service entities in domain A and the final target (file server1), and restriction2 added by *intermediate*1 (printer scheduler) limits the target only to file server1. The final target in domain B, target, needs to verify the fact that intermediate1 provides a printing service in domain A. However, the fact is about the different domain A and target in domain B needs extra information to verify the fact.



(1): $privilege + restricition1$

(2): $privilege + restricition1 + restriction2$

(3): $privilege + restricition1 + restriction2$

Figure 5.2: Delegation across Security Domain Boundary

In order to solve these problems, the SESAME architecture provides an

inter-domain server which can verify seals of PA-Servers of other domains and if necessary translate privileges and restrictions to the local representation. For the purpose of these services, the inter-domain server must support the seal algorithm of PACs generated by the PA-Servers, get the PA-Servers' keys, and know the mapping information of privileges and restrictions between the local domain and other domains. However, this is not always true in a large network environment which consists of many different security domains.

5.4.3 New Mechanism

A more practical solution is that the chained PACs or proxies are translated to a single PAC or proxy which can be verified by the target, and the translation is done in the initiator's or the intermediate's domain. If necessary, the keys or the certificates of the keys issued by a trusted third party may be attached to the PAC.

The refine Privilege Attribute Certificate (refinePAC) service provided by the PA-Server [ISO1993] may be used for the translation from the chained PACs to the single PAC; this service is originally intended to tailor set of privileges and controls the PAC, such as longer validity time, depending on the applications which the initiator wants to use. However, with the information of policies, trusted authorities, supported seal algorithms, privileges and restrictions of the final target, the refinePAC service can generate a single PAC which can be verified by the target, and can solve the problems mentioned in the previous section.

Figure 5.3 shows the translation of the PACs. User's process request a job requiring access to resource in Domain C to Server G with Public Key Certificate (PKC) of the user, request signed by the user, and PAC containing user's privilege, the user belongs to Group S in this case, signed by the authority of Domain A (1). In order to access to the resource with the privilege, Server G requests a new PAC containing the privilege to PA-Server with the PAC of the user. The server authenticates that the request comes from Server G and the PAC is delegated to the server with the restriction attribute in the PAC, and then the server issues a new PAC containing the privilege with restriction of recipient, Server H (2). Server G accesses Server H with its PAC, request, and the new PAC containing the privilege of the original requestor, User X (3).

The key point is the information of the policies, the trusted authorities, the seal algorithms, the privileges and the restrictions which are used for the translation. In the case that the two domains trust each other, the information is held locally and the refinePAC service is enforced to use the seal algorithm, the privileges and the

restrictions supported by the target domain. When both domains do not trust each other, the refinePAC service needs to find a trusted third party which is trusted by the two domains and needs to know seal algorithms, privileges and restrictions which are commonly supported by both domains. The information can be retrieved via a directory service; each security domain needs to announce the following information:

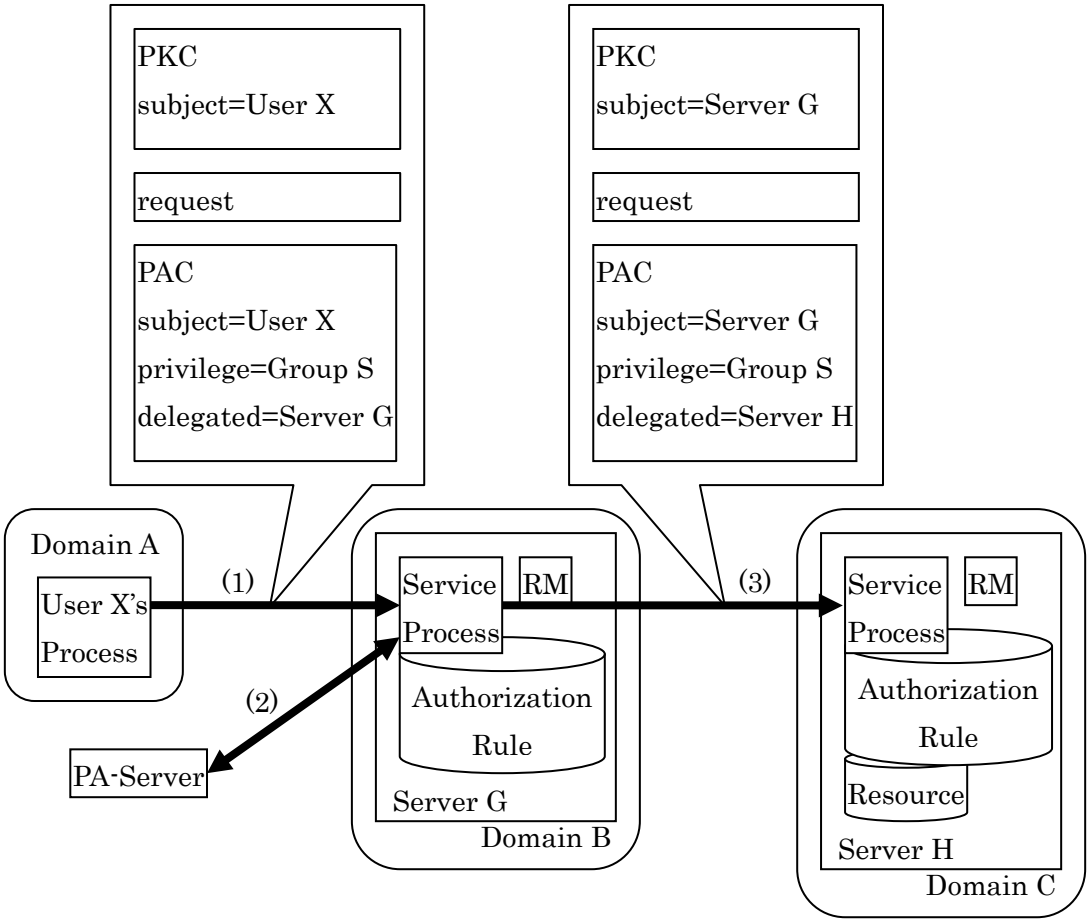


Figure 5.3: Delegation with Single PAC

- Policy

  The security policy identifiers which specify policies adopted by the domain with qualifier information pertaining to the policies; the syntax of the information is given in the amendment to X.509 [ISO1995]. A security policy may specify acceptable use, cryptographic algorithms, user certification procedures or operational matters such as validity time length of certificates, etc. A domain may also specify prohibited policies which the domain cannot accept. The PA-Server needs to check that there is no contradiction between its policies and those of the

target domains.

- Authority

  A set of authorities trusted by the domain; the authority may specify the security policies which should be adopted by domains trusting the authority. The PA-Server of the initiator or the intermediate needs to find an authority which is commonly trusted by the initiator / intermediate domain and the target domain.

- Seal Algorithms

  A set of seal algorithms which the domain supports; the PA-Server uses an algorithm which is commonly supported by the PA-Server and the target domain. In most case the algorithms specified by the commonly trusted authority will be used. A security policy of a domain may specify the precedence of algorithms.

- Privilege and Restriction

  A set of privileges and restrictions which the domain supports and optionally mapping information between standard ones and ones defined locally; the PA-Server translates local privileges and restrictions to the remote ones. In many cases, the privileges and the restrictions defined by the commonly trusted authority or adopted policies will be used.

Note that the above information may need to be protected; with forged information, the refinePAC service generates a PAC which cannot be accepted in the target domain, and this leads to denial of services. This method has the following advantage and disadvantage:

- Verification Cost

  The verification cost of privileges and restrictions at the target side is reduced because the chained PACs and proxies have already been deleted at the initiator's side and the target can check authorization immediately after the verification of the single PAC. Instead of this, the cost of issuing the single PAC increases at the initiator's side. However, normally this is not a problem because requests coming from many initiators make the target system's load heavier than the initiator's in general and it is desirable to distribute and reduce the cost of the verification. This is also true about the cost of the mapping of the privileges and the restrictions.

- Trace Information

  The translation of the chained PACs to a single PAC makes the trace of the delegation path impossible. However, adding a new parameter, which includes the initiator and the intermediate(s), filled by the PA-Server to the PAC makes possible

sary information for auditing and charging.

The ECMA standard [ECMA1989] has proposed another inter-domain service mechanism; the initiator inter-domain service translates local privileges to standard ones and signs the PAC with the inter-domain service key, next the PAC is passed to an inter-domain server of a trusted third party. The PAC is verified and signed by the authority of the trusted third party, and send back to the initiator through the inter-domain server. Finally, the re-signed PAC is passed from the initiator to the target and the inter-domain server in the target domain, and is verified and translated to local privileges and restrictions.

The differences between the ECMA's mechanism and the proposed one are the route of PACs and certificates included in the PAC. While in the ECMA mechanism a PAC is routed from the inter-domain server, the trusted third party, the inter-domain server, the initiator and finally to the target domain, in the proposed mechanism the PAC is directly passed to the target. The seal generated by the PA-Server in the initiator domain can be verified by the target because the PA-Server adds the key certificate of the server issued by the authority of the trusted third domain to the PAC. The privileges and restrictions contained in the PAC are acceptable in the target domain.

With the combination authorization with PAC and CAP, and privilege delegation with single PAC, the problems stated in Section 5.1 are solved. Formulation of restriction and authorization rule, the first and third problem, are realized with classification of attributes in the PAC and CAP in described in Section 5.3; six classes are defined and the decision method, which class attribute is compared against which class attribute with comparing function defined by the attribute syntax, is also defined. The second problem, chained PACs, is also solved by translation to single PAC with PA-Server described above.

5.5 Implementation of ACDF

The PAC/CAP authorization schema has been implemented on a WAIS server; a generic ACDF has been implemented and the WAIS protocol (the initialization phase and the document retrieval phase) has been extended to support the new authorization mechanism. Each of them is described in the following sections. As for the new delegation mechanism, the new refinePAC service which translates chained PACs into a single PAC will be implemented later.

5.5.1 Access Control Decision Function

The function takes six sequences of security attributes as arguments, namely, attributes of the privilege, positive and negative restriction, condition, exception and context classes, and returns OK which means the access is allowed, NG indicating the access denied and UNKNOWN when unrecognized attributes are given.

The authorization rule is managed by security attribute tables; the following table shows how a condition attribute class is configured:

category:	IncludedSEOFPrintablestring:	category:prv
clearness:	smallerINTEGER:	clearance:prv
accesstype:	IncludeSETOFInteger:	accesstype:ctx
subjectAddress:	IncludeIPAddress:	address:ctx
permittedAccesstime:	IncludeTime:	accesstime:ctx
minimalAithenticatedLevel:	SmallerINTEGER:	authenticatedLevel:ctx

Each line specifies a condition security attribute and consists of three tuples: the attribute name, the attribute syntax which defines value type and the matching rule of the values, and the attribute name of the privilege (indicated with :prv) or context (:ctx) class which is compared with the condition attribute. For example, the last line specifies that the “minimal authentication level” condition attribute must be smaller than the “authentication level” attribute value of the context. The authorization rules for exception, positive and negative restriction classes are configured in the same manner.

5.5.2 Initialization Phase

For the purpose of simple description of the extend protocol, notations listed in Table 5.1 are used in the following.

(1) Initialization Request

The subject’s privileges and the restrictions are transmitted in the initialization phase in the form of the PAC as well as authentication information and a secret session key; all information is carried in the idAuthentication parameter of the InitializeRequest of the WAIS protocol.

$$C \rightarrow S : \{AuthInfo, PAC, KeyPack\}$$

where

$$AuthInfo = \{S, random, time\}_{prvKey(C)}$$

$$PAC = \{serialNumber, P, R, id, PS\}_{prvKey(PS)}$$

$$KeyPack = \{SK, S, time, random\}^{pubKey(S)}$$

Table: 5.1 Notations

Notation	Description
C	Client
S	WAIS server
PS	PA-server
$AuthInfo$	Authentication information
$KeyPack$	Session key package
$\{I\}_k$	Information I singed with key k
$\{I\}^k$	Information I encrypted with key k
$pubKey(X)$	Public key of enitity X
$prvKey(X)$	Private key of enitity X
P	Privileges
R	Restrictions
id	Audit identifier and charging identifier
SK	Session key
dek	Data (document) encryption key
$MD(I)$	Message digest of information I

The authentication information ($AuthInfo$) is same as the bind-token of the directory access protocol [CCITT1988]; it includes the intended recipient (S), a random number and the current time, and sealed with the client's private key ($prvKey(C)$) of the RSA encryption algorithm [RSA1993a]. The client's public key certificate [CCITT1988] might be attached to the authentication information.

The PAC contains a serial number, privileges P, restrictions R, an authority name PS and identifiers for auditing and charging id. The PAC is sealed with the PA-Server's private key $prvKey(PS)$ and distributed in an off-line manner.

The format is different from one defined in [ECMA1989]; the validity time is omitted because it is included in the positive restriction, the restriction type is changed because of simplicity and direct comparison against condition and context security attributes, and the contained PAC parameter is omitted because the PAC is always packed into a single PAC described in the previous chapter and the parameter is not used.

A randomly generated session key SK, the intended recipient S, time and a random number are encrypted with the server's public key $pubKey(S)$ and packed in the session key package $KeyPack$.

(2) Initialization Response

The server verifies the authentication information with the client's public key which might be stored in a local cache or retrieved from the public key certificate attached to the authentication information. If the verification succeeds, the server checks whether the included time $time$ is enough to close to the current time of the server, and the random number $random$ was not used before. After verification of the PAC, the server checks the positive restrictions of the PAC which might restrict the subject. In this case the server needs to check the authenticated client equal to the subject.

The server decrypts the session key package with the server's private key, confirms that the included name is same as the server's name, and checks the included time and random in the same way as one of the authentication information. The session key is used to realize integrity and confidentiality of documents during transmission from the server to the client.

The server sends back the normal WAIS initialization response in the current version. For the mutual authentication between the server and the client, the server needs to send back the authentication information of itself in the idAuthentication parameter of the InitializationResponse:

$$S \rightarrow C : \{C, random, time\}_{prvKey(S)}$$

where the random ($random$) equals to the random in the session key package. With the verification of the information, the client can authenticate the server and make sure that the session key package is correctly decrypted and the session key is shared between the two entities.

5.5.3 Document Retrieval Phase

Each time the client requests to retrieve a document, the ACDF is called with arguments of the six classes of the security attribute, namely, the privileges and the restrictions got from the PAC, context information managed by the server, the conditions and the exceptions of the associated access control class; each document is tagged with an access control class identifier and each access control class is associated with a CAP, a sequence of condition and exception security attributes.

For the purpose of confidentiality of the retrieved document, a randomly generated data encryption key dek is used for the encipherment of each document, and send with the enciphered document after it is encrypted with the session key. For the sake of integrity, a message digest of the document is encrypted with the session key and send with the enciphered document. Since only the client and the server share the session key, the client can decrypt the enciphered document and check integrity. These three components is handled as a single document in the protocol.

$$S \rightarrow C : \{dek\}^{SK}, \{document\}^{dek}, \{MD(document)\}^{SK}$$

Currently the DES-CBC algorithm [NIST1977] is used for the two services and the MD5 message digest algorithm [RR1992] in order to generate message digests of documents. In view of security, the retrieval request should be authenticated; the request should be sealed with the shared session key by the client. However, there is no appropriate parameter it the WAIS protocol, the authentication of the request is not implemented.

5.6 Conclusions

In this chapter requirements of authorization especially for information servers running on an organizational scale and a large scale network are enumerated, the problems of the ACL-based authorization are pointed out, and the PAC/CAP authorization scheme has been proposed which has several advantages when it applied to such information servers. Next delegation problems have been pointed out and a solution using translation of chained PACs into a single PAC with help a directory service has been proposed. Finally an implementation of a WAIS server and a client has been presented. Currently the PAC is signed by the authority with an asymmetric encryption algorithm and distributed in an off-line manner; in later versions an on-line distribution of the

PAC supported by the new refinePAC service will be introduced. Auditing is a new frontier of security which the authors have not addressed yet. The current ACDF library records only what subject with what auditing identifier is permitted or rejected to access to which object. This auditing trail might not be enough because the context information is not recorded which cannot be traced after the decision. The authors will examine what kind of information is necessary during authentication, authorization and real processing of requests from the subject, and implement in later version.

With the advent of the grid computing, authorization in multiple domains becomes a real problem. In UNICORE delegation model [SD2004], an end user signs job and sub-job as en endorser, request them to a server. The server transfer the sub-job to another server as a cosigner, and then the sub-job is executed with the privilege of the endorser, the end user. Moreover, if the server is explicitly trusted, the server can play a role as the endorser on behalf of the end user. Comparing the UNICORE model, the privilege of an end user is transferred in the PAC in the proposed delegation scheme. The proposed scheme has more flexibility to represent the end user's privilege, but it is more complicated, and processing costs much.

The PERMIS Project [CD2002a, CD2002b] realized the hierarchical Role-Based Access Control (RBAC) based on the X.509 Attribute Certificate (AC) [ITU2001], implemented ACDF with JAVA API. The implementation is used in reality [CD2002b], while the ACDF of the proposed scheme is a just prototype. Both of AC and PAC convey user's privilege in a certificate, and it is used by ACDF. The distinguishing characteristic of PERMIS is the hierarchical RBAC model; with the model the role specification is more compact and then understandable. The proposed scheme is more flexible when delegation of privilege from a user to a service entity is required during job processing in distributed computing environment, while delegation of role assignment is supported in PERMIS, which is not used during job processing.

All of the PKC, PAC, and AC are protected with signature based on public key cryptography. However, it is also possible to protect the certificates with secret key cryptography; authentication and authorization scheme with secret key and privilege attribute certificates [SY1996b, SY1997b] was realized and used in commercial products. The reasons why the standardized certificate was not adopted were firstly the slow speed of the public key algorithm implemented on computers of that time, and secondly licensing cost of the algorithm.

Chapter 6

Prevention of Virus Infection and Secret Leakage with Secure OS and Virtual Machine

6.1 Introduction

In this chapter, integration of e-mail clients which are separated in data isolation system is described. The system is introduced as a countermeasure for unknown virus, but a user of the system should change PC operations from the current usage. The integration reduces burden of the user, because the user does not need to be aware of separation of data.

The top two IT security threats are virus infection and secret leakage including PC theft [GL2006, SYM2007, MCA2007]. The best practice to prevent virus infection is to use virus protection software and install security patch. However, the practice is becoming less effective, because of the following two reasons: The first reason is zero-day attack [SYM2007]; the attack code appears soon after vulnerability is announced, for example the attack code of MS07-002 [MS2007] appeared three hours later after the patch was released. As a result, virus definition file cannot be in time. The second reason is targeted attack [FS2006]; while existing attack codes aim to be spread over many victims, the target of the new attack is very limited, for example a single organization or few people. As a result, there is less chance to detect the target attack and the virus definition file may not be issued.

A solution against the virus threat is to utilize secure OS [ARGUS2001, LP2001]. The OS supports the Mandatory Access Control (MAC) [DD1983], and the damage of attack to vulnerability of application is limited only to the application; the attack code cannot access file nor execute process which are not permitted in security policy, even if the code gets the administrator privilege. The secure OS is used mainly for server, but not client PC, because of management of the security policy; it is difficult for end user or system administrator to configure the security policy specifying which process is permitted to access to which resources with what kind of operations.

Another solution is behavior based virus detection [CM2005]. Virus has some specific behavior; some of virus code is encrypted to bypass the virus protection system, and decryption of code is one of features of the virus. Another virus sends many e-mail

messages of its own copy. The new technology watches such behavior of virus and detects the virus, but the new virus detection may miss targeted attack, because the virus targets specific organization or information, and the virus may be tuned so as not to be detected by such virus protection software.

On the other hand, more serious IT security threat is leakage of secret information or secret leakage [MCA2007]. The main reason of secret leakage is lost of PC or storage media, but other reason is intentional leakage by authorized user and exposition to the Internet by virus.

Solutions against secret leakage are file encryption and prohibition of portable storage media/printer. The Windows OS supports file encryption, and its security policy can enforce to stop use of USB memory. However, these solutions are not effective for intentional leakage through e-mail or HTTP by authorized user; it is possible to stop sending e-mail outside or posting via HTTP, however, this is not practical for commercial organizations.

NetTop [HP2004, MR2000] is a countermeasure of the two threats; it is designed for intelligence community, and the goal is data isolation. User of NetTop accesses classified information of multiple categories and operates multiple workstations which are integrated into a single PC with Trusted Linux and Virtual Machine (VM). The workstations are separated virtually at physical level, so the threats do not happen. However, the user should always be aware that which workstation she/he is operating and needs to switch the two workstations. This is acceptable for users of intelligence community, but it is very troublesome for office workers of commercial companies.

"Windows Vault" is proposed as system isolation which usage is as same as a normal Windows as possible. The user operates a safe workstation isolated from the Internet, but she/he can access the external information that comes from the Internet on the safe workstation without threats of virus infection or secret leakage. The word 'Vault' means a room with thick walls and strong doors where valuables can be kept safely; Windows Vault is a vault running Windows, that is, a Windows workstation guarded by secure OS and gateways which establish secure data exchange between the isolated workstation and the external environment including the Internet.

In this chapter, the architecture of Windows Vault is described in Section 6.2, evaluation of performance, security and usages in Section 6.3 through 6.5, compares with the previous works in Section 6.6, and concluded in Section 6.7.

6.2 Concepts of Windows Vault

The principal of Windows Vault is very simple; data is divided into two categories, safe secret and unsafe non-secret, and the later includes information on the Internet and may contain virus. Windows Vault processes the two data categories with two virtual workstations; Internal Workstation for safe secret, and External Workstation for unsafe non-secret, and the two workstations are integrated into a single physical PC with use of VM and secure OS. Network is also divided; Internal Workstation is connected to Internal Network and External Workstation to External Network including the Internet.

The above architecture realizes very high level security, as far as user processes the two categories in completely separated manner. But such use is not realistic. While main task of user of commercial company is processed on Internal Workstation, the user also needs to access the Internet and utilize information of the Internet as part of secret; text on Web and spread sheet data attached to e-mail from business partner are examples of such information. It is also desirable to use a single e-mail client; the user does not want to use two clients on Internal and External Workstations, because it is different from the current e-mail client usage. As for web browser, the other most used network application, it is normal that user operates multiple browser windows, and it is desirable that the user can operate browser window accessing a site on the Internet in the same operation of the window on Internal Network. As a result, the following functions are required with security guaranteed form:

- Data import: data is imported from External Workstation to Internal Workstation.
- Mail retrieval and sending on Internal Workstation: user operates e-mail client on Internal Workstation, retrieves and sends messages with the client from/to Internal and External Networks.
- Browsing Internet sites from Internal Workstation: user operates web browser on External Workstation from Internal Workstation.

With the above functions, the user needs to use Internal Workstation only and she/he can process information on External Workstation on Internal Workstation. Four gateways connect the two workstations and realize secure channel between the two workstations for e-mail and copy & paste operation. In the following, the overall architecture, platform OS, and four gateways are described.

6.3 Architecture of Windows Vault

6.3.1 Overall Architecture

The overall architecture of Windows Vault is shown in Figure 6.1. Platform OS is the base of security, and the Security-Enhanced Linux (SELinux) [LP2001] is adopted, which supports the MAC based on the Type Enforcement model. Each of Internal and External Workstations consists of VM, Windows OS, and applications.

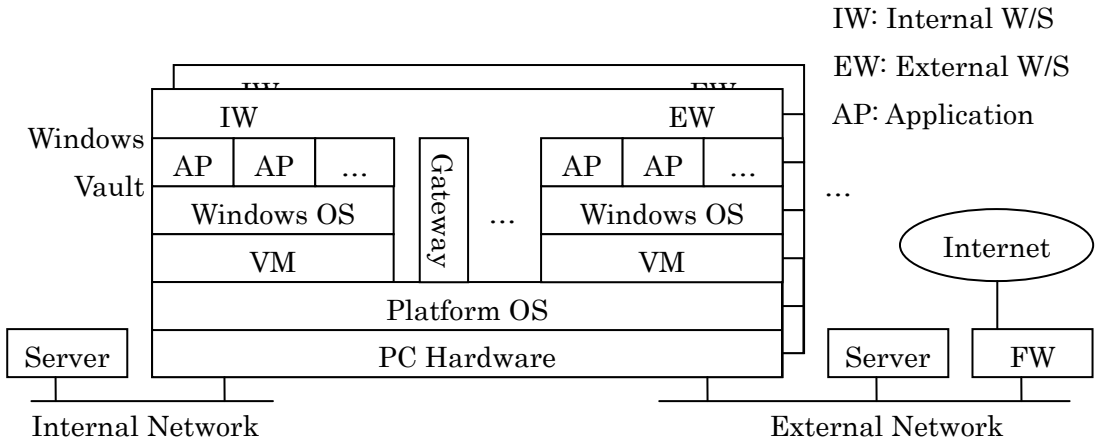


Figure 6.1: Architecture of Windows Vault

Two LANs, Internal and External Networks, are connected to Platform OS. It is assumed that the network devices connected to Internal Network are managed, that is, only identified and authorized devices are connected to the network, and virus infected PCs are not connected.

The virtual networks on Platform OS are configured as shown in Figure 6.2; Internal Workstation is connected to Internal Network and the virtual Internal Network, so on External Workstation, and each gateway is connected to the virtual networks and External Network. Data exchanged between Internal and External Workstations is limited only through the gateways by the configuration of Platform OS.

6.3.2 Platform OS

In order to prevent user from changing configuration of Platform OS, direct access to Platform OS must be prohibited. The following configuration realizes this:

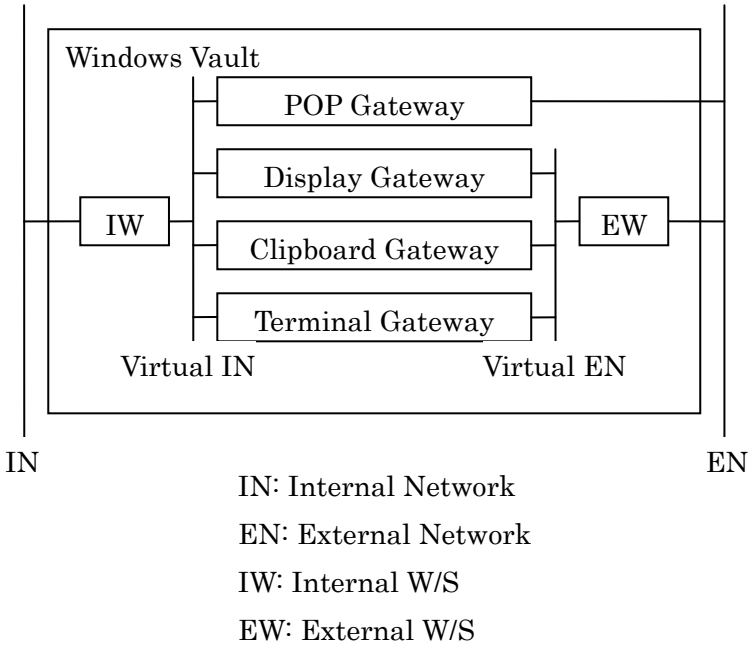


Figure 6.2: Gateways on Virtual Internal Networks

- The default run level is changed to level 4.
- The "init" process starts services required to manage Platform OS, for example "syslogd," and does not start "getty," nor trap ctl-alt-del.
- The starting script kicks the following programs: the X window system display server, four gateways, two VMs, and screen lock program.
- The shutdown process runs after both the VMs end.

The security policy of Platform OS is configured as shown in Figure 6.3. The init process launches the xinit command, and the command starts the X window system display server, VMs and gateways. Each VM accesses three files, log, configuration, and virtual disk image, and each of the files is assigned a different type of SELinux. The access kind of each type is minimal, for example, wv_int_log_t, the type of log file of Internal Workstation, is written only by wv_int_t, the domain of Internal Workstation.

6.3.3 Gateways

The first request, data import from External Workstation to Internal Workstation without virus infection, is realized by Clipboard Gateway, which works as follows:

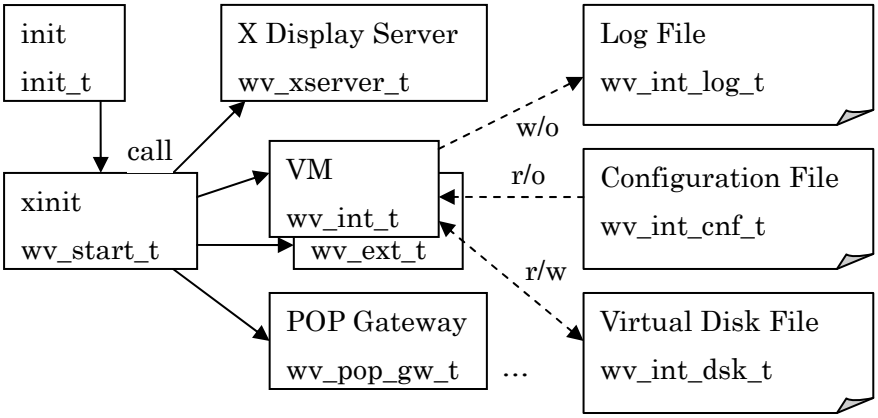


Figure 6.3: Security Policy of Platform OS

- The clipboard watch agent on External Workstation transmits object on the clipboard to the gateway when the user copies object.
- The gateway checks the object type, and transmits the object if the type is not file. Otherwise the gateway does not transmit.
- The agent copies the object to the clipboard of Internal Workstation.

In order to realize part of the second requirement, e-mail retrieval from External Network, the e-mail client on Internal Workstation accesses two POP servers on Internal and External Networks, and the client accesses the later through POP Gateway, which encapsulates attached files. The encapsulation and decapsulation processes are shown in Figure 6.4.

POP gateway encrypts each attached file with a randomly generated AES key, encrypts the random key with an encryption key of RSA, and signs the encrypted random key and file with a signature key of RSA. When the user opens the attached file received from External Network, it is sent to Display Gateway, the gateway checks the signature with the verification key corresponding to the signature key, forwards the encrypted key and file to the display agent on External Workstation, and then the agent decrypts the file and displays it. The e-mail from Internal Network is opened normally on Internal Workstation.

Attached Files from the Internet are opened in External Workstation and they are accessed safely from Internal Workstation through Terminal Gateway. Actually the gateway is a remote access client or terminal client running on a remote access server or terminal server; as shown in Figure 6.5, combination of two remote accesses, one from Internal Workstation to Terminal Gateway and the other from the gateway to External

Workstation, realized a remote access from Internal Workstation to External Workstation.

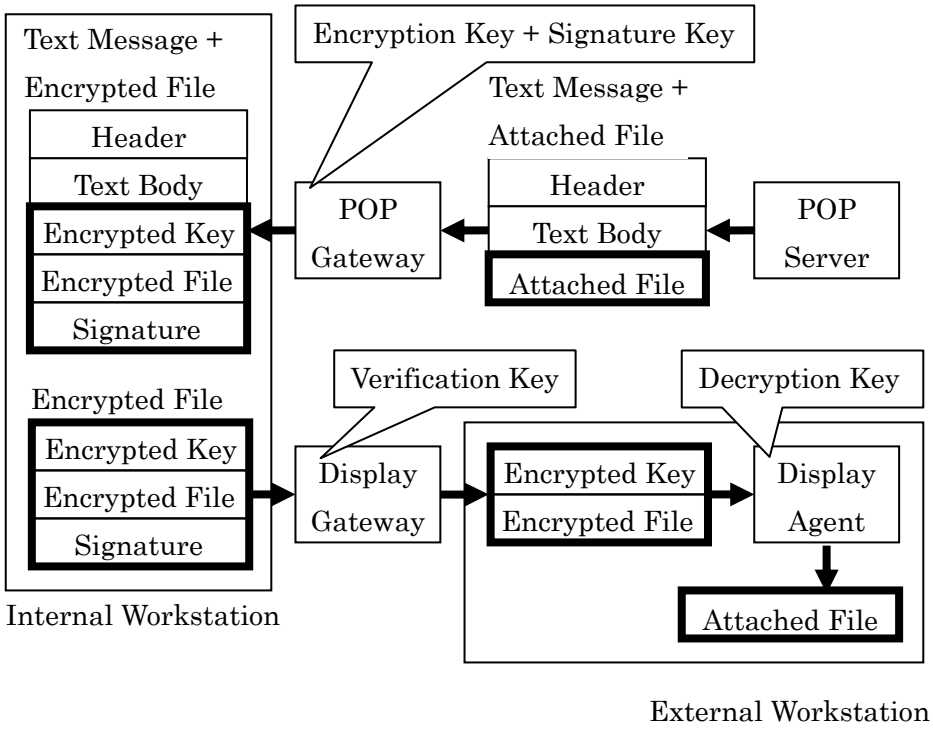


Figure 6.4: Encapsulation and Decapsulation of Attached File

With use of Terminal and Display Gateways, the user can open an attached file of e-mail from the Internet with the same operation of opening attached file from Internal Network; the file is automatically displayed in the terminal client on Internal Workstation, and it looks almost the same as file opened in Internal Workstation locally as shown in Figure 6.6.

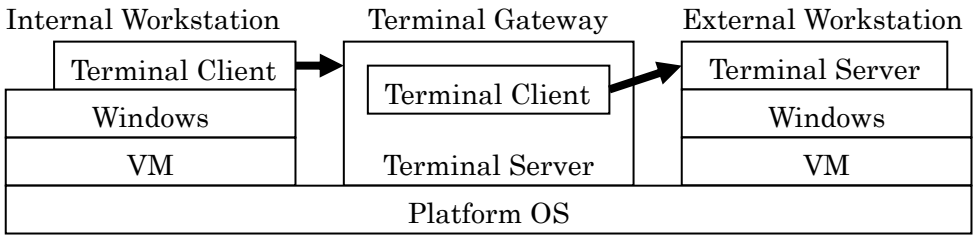


Figure 6.5: Terminal Gateway as Combination of Two Remote Accesses

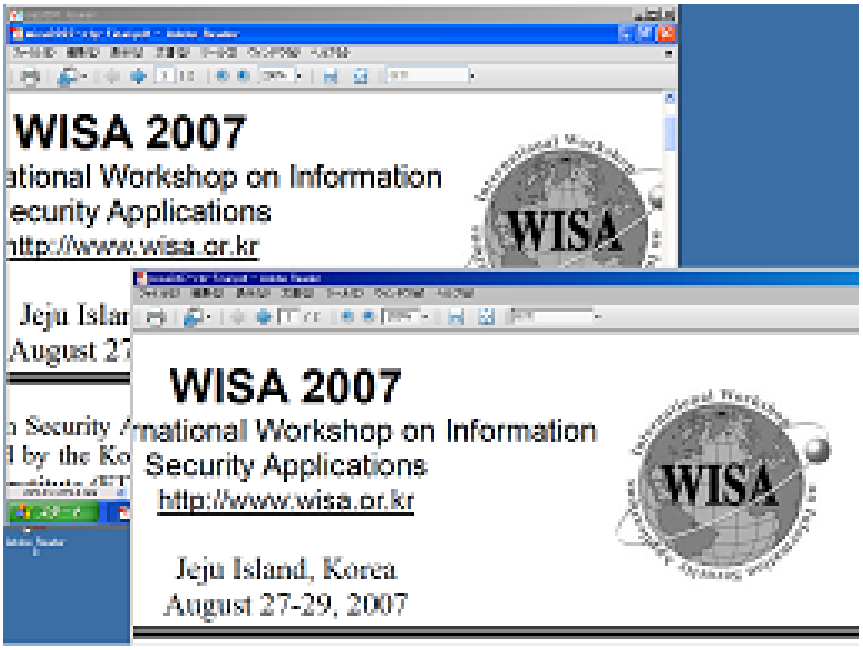


Figure 6.6: The Same Files Opened in Internal Workstation (right) and External Workstation (left)

6.4 Performance Evaluation

The authors have implemented a prototype of Windows Vault and measured performance in the environment shown in Table 6.1 with the benchmark program, CrystalMark 2004R2 [CDW], and result is shown in Table 6.2. The column M is the result of the mean score of Internal and External Workstations measured simultaneously, S is one of Internal Workstation, and W is a normal Windows PC with Intel Core Solo (1.66GHz).

Table 6.1: Environment

Item	Description
Platform OS	Cent OS release 5
VM	VMware Workstation 5.5.2
Windows	Windows XP Pro. + SP2
CPU	Intel Core 2 Duo (2.16GHz)
Memory	2GB

Table 6.2: CrystalMark Results

Item	M	S	W
Integer	8,361	8,383	3,901
Float	9,511	9,465	4,584
Memory	10,447	10,722	4,764
HD	11,457	11,985	4,485
GDI	1,979	2,205	1,426

Comparing with the normal Windows, the performance of Internal Workstation and External Workstation is better, and each of the workstations has shown enough performance as a Windows PC.

The performance of retrieving e-mail and opening attached file is shown in Table 6.3. The column N is the time without POP Gateway, and the column P is the case through the gateway. The overhead is about 0-90%. The column D is the transmission and decryption time of encrypted file through Display Gateway. There is overhead of encryption and decryption, however, it is not so heavy to give impact on usability.

Table 6.3: Performance of POP Gateway and Display Process

File Size (B)	Msg. Size (B)	N (sec)	P (sec)	P/N	D (sec)
10K	15K	0.198	0.198	1.00	0.018
345K	467K	0.206	0.280	1.36	0.438
3,262K	4,406K	0.522	0.977	1.87	1.954

6.5 Security Considerations

6.5.1 Attacks from External Workstation/Network to Internal Workstation

Internal Workstation is not directly connected to External Network, but there are four routes of attack from External Network to Internal Workstation. The first route is via e-mail; the e-mail client receives messages from External Network through POP Gateway which encrypts attached files, and the files cannot be opened on Internal Workstation. As a result, there is no possibility to infect virus via e-mail attached file.

Virus might be included in header or text body, so the gateway should check character code and line length, and sanitize if they do not meet the protocol specification.

The second is via VM; External Workstation may be infected with virus which attacks the base VM, and such virus may attack Internal Workstation. But the MAC of Platform OS does not allow access between Internal and External Workstations, and such attack cannot happen.

The third is via Clipboard Gateway. Normally user sees and selects an object, and copies it to the clipboard, so the possibility of virus in clipboard object, which is not a file, is considered to be low, but the object might contain virus code. In order to avoid such possibility, the following object check functions of the gateway are useful:

- Plain text only: A clipboard object consists of type and data, and the gateway checks the type and only text object is transmitted to Internal Workstation. The size of text data and character code are also be checked.
- Strictly defined data: If the object data type is strictly defined, it is possible for the gateway to check the clipboard object meets the definition and does not contain virus.

The fourth route is via Terminal Gateway; basically the gateway transmits the keyboard and mouse events from Internal Workstation to External Workstation and graphical screen data from External Workstation to Internal Workstation, so virus code cannot come into Internal Workstation. Normally a remote access protocol supports clipboard sharing, however, Terminal Gateway kills the function, and virus infection through Terminal Gateway does not happen.

6.5.2 Attacks by User

User may try to leak a secret file on Internal Workstation to External Workstation through Display Gateway, but the file is not forwarded to External Workstation, because the signature verification at the gateway fails. Consequently, there is no secret leakage of file created on Internal Workstation to External Workstation or Network through Display Gateway. The information flow of POP Gateway and Clipboard Gateway is only from External Workstation to Internal Workstation, therefore secret leakage does not happen. Terminal Gateway transmits display image, mouse and keyboard events, and secret leakage from Internal Workstation to External Workstation cannot happen except that a malicious user leaks secret text by typing keyboard.

The user can change consoles of character terminals or X Window by typing ctl-alt-function key, however, Platform OS is configured as no getty and screen lock program is running on the X Window console, so the user can only access Internal and External Workstations. As a result, normal office worker cannot access Platform OS, nor change its configuration.

But the user who has knowledge of Linux management can access Platform OS by trapping the boot process or direct access to hard disk. Possible solutions are change of the init process program, physical lock of the PC hardware or use of the Trusted Platform Module (TPM) [TCG2007]. The TPM is a chip on a PC motherboard and calculates hash values of software components. On request from a remote challenger, the TPM returns the hash value with signature generated within the chip. With this attestation process, the remote challenger can authenticate PC hardware and verify software integrity of Platform OS, and as a result, it is possible to detect physical attacks such as replace of PC hardware, workstations and gateways. The TPM is also used as a key storage; it is possible to encrypt the virtual disk images of the two workstations and decrypt only on the specific PC hardware that has the TPM storing the decryption key, and this countermeasure makes the attack of direct access to hard disk useless.

A few services, such as system logging, are running on Platform OS, and it might be possible to attack such services. However, each service is given a domain and separated from the other processes by the MAC, so Internal Workstation cannot be attacked through a service even if there is vulnerability of the service.

6.5.3 Vulnerability of Gateways and Enhancements

If POP Gateway has vulnerability, attacker may get the control of the gateway and can inject virus to Internal Workstation or steal secret from Internal Workstation. It is the same as Clipboard and Display Gateways. The MAC of Platform OS cannot cover the weakness of the three gateways, and the security quality of the gateways is very important. However, it is possible to enhance the security by dividing function of each gateway as follows:

Display Gateway has three functions; firstly it receives encrypted and signed file from Internal Workstation, secondly verifies the signature and strips it, and finally sends the file to External Workstation. The three functions can be realized by three processes of different domains, receive process of r_display_t domain, verify process of v_display_t domain, and send process of s_display_t domain. Data between processes is

passed via files of different types; the receive process receives file from Internal Workstation and saves it of rv_display_t type, the verify process verifies and strips the signature and saves data as file of vs_display_t type, and the send process sends it to External Workstation. The permitted operations between the domains and types are shown in Figure 6.7.

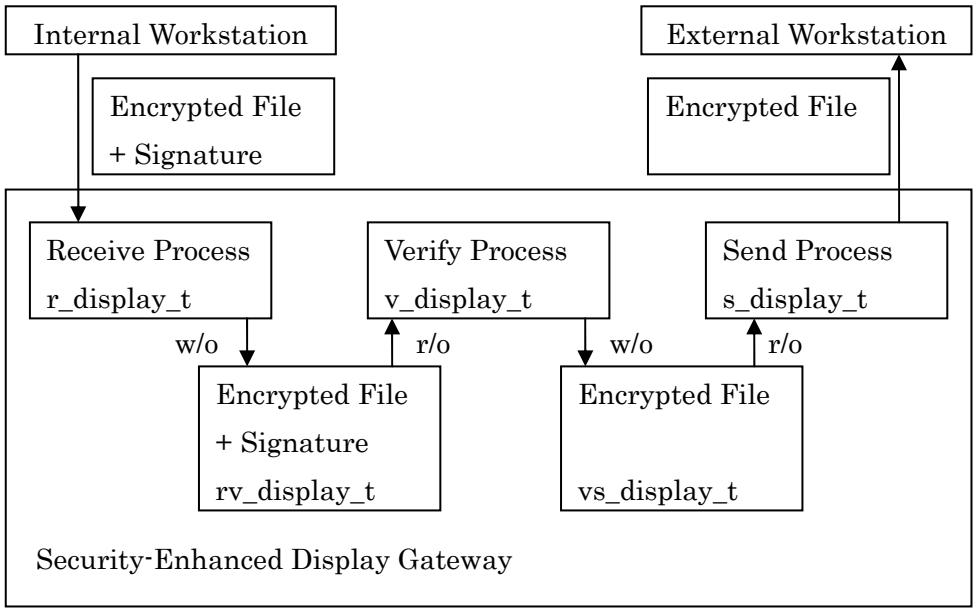


Figure 6.7: Domains and Types of Security-Enhanced Display Gateway

Comparing the original implementation of the gateway, the new gateway is more secure, because of the following reasons:

- An attacker, which might be the user, must exploit the vulnerability through file, which is enforced by the MAC, and the attack through file is more difficult than one through TCP/IP communication channel, because the former is not interactive and less measures of attack.
- The order of attack is fixed; first the reception process, next the verification process, then the send process, and there is no other pass of attack, because the other pass is prohibited by the MAC, and the attacker has less means to attack.
- Each process realizes one function and the code is simpler, so that it is more secure than process supporting multiple functions.
- Even if an attacker succeeds to exploit all the processes, the information flow is limited from Internal Workstation to External Workstation by the MAC, and there

is no chance virus infection of Internal Workstation.

The situation of Clipboard Gateway is very similar to Display Gateway. The gateway has three functions; firstly it receives clipboard object from External Workstation, secondly verifies the type object, and finally sends the object to Internal Workstation. The security of the gateway can be enhanced in the same way of Display Gateway.

Since the POP is an interactive protocol, the situation of POP Gateway is different from the other two gateways, but the basic idea is the same; the first process receives messages as a POP client, the next encrypts and signs the attached files, and the third behaves as a POP server and sends the messages to Internal Workstation.

While the fail of security of the three gateways before the enhancements leads to catalysis, the channel through Terminal Gateway is safe even if there is vulnerability in the gateway. The gateway consists of two applications, terminal server and terminal client which domains are different. As a result, if an attacker gets control of one application, the attacker cannot get one of the other. Even if one application has vulnerability and an attack code succeeds to transmit any type of data, the other application transmits only graphical display data, keyboard and mouse events, so that the security of the gateway is guaranteed.

6.5.4 Another Data Category: Unsafe Secret

Current Windows Vault processes two data categories, safe secret and unsafe non-secret, but there is another category data, unsafe secret. Online banking is a typical example; account number, password, balance sheet are secret, but the web page data sent from the server may contain virus. Another typical example is e-mail message sent from a business partner; an attached file contains business secret but the file might be infected with virus.

The current Windows Vault processes such data as unsafe non-secret, because it comes from External Network, and there is a risk that such secret data is stolen from External Workstation. A solution is the third type workstation, which can access only trusted web sites through encrypted and authenticated channel over the Internet. An attached file of e-mail is transmitted through Display Gateway and opened on this workstation; even if the workstation is infected with virus, secret cannot be leaked out to the Internet, because the workstation is connected only the trusted web sites.

6.6 Usability of Network Applications

In the following, usability of sending e-mail and web browsing is described.

6.6.1 Sending Message to External Network

In the current implementation, user can receive e-mail message from External Network with e-mail client on Internal Workstation, but cannot reply to the message, because Internal Workstation cannot access External Network, and this leads to inconvenience. It is also true that the user cannot send a new message from Internal Workstation to External Network. As far as user sends from Internal Workstation to External Network, the only solution is encryption; all messages from internal to trusted external recipient are encrypted by the fifth gateway, SMTP gateway, which is connected to the Virtual Internal Network and External Network, and encrypts all received e-mail from Internal Workstation.

A solution to reply to message from External Network is to add the original whole message as an attached file to the message; POP gateway encrypts and signs the whole message, and then adds it as the last attached file. When user wants to reply to the message, the user selects the last attached file to open, and then the file is sent to External Workstation through Display Gateway, an e-mail client opens the file and displays the original message, and then the user replies with normal operation through Terminal Gateway.

As for a new message to External Network, the user needs to send it with the e-mail client on External Network. However, by sending carbon copy to the user own account on Internal Workstation, the user can access the new message on Internal Workstation.

6.6.2 Web Browsing

With click of links, user can brows web pages without consciousness of the page location. Windows Vault divides OS and network into internal and external, so the user needs to be conscious of which network the accessing site belongs to, and changes browsers on Internal and External Workstations. This is big change of usage of web browser.

In order to realize ‘smooth browsing from internal to external,’ it is better that with click of link to external page on internal page, the external page is displayed on Internal Workstation. This function can be realized with HTTP Gateway which calls

web browser on External Workstation. The gateway is connected to the virtual External Network and Internal Network, and behaves as follows:

- The web browser on Internal Workstation accesses the gateway according to the proxy configuration of the browser.
- The gateway returns an error page to the browser, and sends the requested URL to the HTTP agent on External Workstation.
- The agent directs browser on External Workstation to access the URL.
- The browser accesses the page of the URL, displays the external page, and user can access the page from Internal Workstation through Terminal Gateway.

With the gateway, the user can smoothly brows from a page on Internal Network to a page on External Network in the same as the current operation. The reverse direction browsing is also possible, but it needs to sanitize the URL in external page, since virus may be contained it the URL.

6.7 Related Works

NetTop also consists of Trusted Linux and VMs, and Windows' on VMs exchange data via 'Regrade Server' with explicit user authorization [MR2000]. The user of NetTop is enforced to use two Windows OSs, two mail clients, two documentation tools, etc., and this is a burden for office workers of commercial companies. On the other hand, the user of Windows Vault accesses only Internal Workstation basically; the user can receive and read text body of e-mail from the Internet, open and read attached files with the same operation as the normal Windows. It is also true that the user can access web pages on the Internet from Internal Workstation through HTTP and Terminal Gateways. This is convenient for users who do not aware of multi-category security.

The Trusted Virtual Domains framework [GJ2005] is a kind of system isolation based on Trusted Platform Module [TCG2007]. The goal of the framework is to establish secure communication channels between software components with the integrity assurance of the other components. The framework also utilizes multiple VMs and software components of different domains running on a hardware platform. There is a secure communication channel between the software components of the same domain, but no communication between those of different domains. However, Windows Vault is focusing the air gap between the different domains or workstations, and has established secure user data exchange between the two workstations.

VIRTUS [IH2006] is a new processor virtualization architecture for security-oriented next-generation mobile terminals. It creates OS instances, called domains, for pre-installed applications and downloaded native applications. VIRTUS supports inter-domain communications, but it does not clearly specify its security. On the other hand, in Windows Vault the communication between Internal Workstation and External Workstation/External Network is designed carefully not to cause virus infection nor secret leakage.

6.8 Conclusions

In this chapter, Windows Vault is described; it consists of two Windows workstations, one for safe secret and the other for unsafe non-secret. The two workstations are integrated into a single PC with use of VM and secure OS, and connected securely by gateways. These gateways transmit data between the two workstations without virus infection of Internal Workstation or secret leakage from Internal Workstation to External Workstation. Comparing with the existing data isolation system, the proposed system realizes the same security without change of current user operations of e-mail or awareness of multi-category security of intelligence community.

Windows Vault which contains only Clipboard gateway is released as a commercial product from Hitachi Software. Later release will contain the other gateway to increase usability as described in this chapter.

A remaining problem is countermeasure against vulnerability of the web site such as cross site scripting or cross site request forgery; this is a problem of web site, not client side, however it is desirable this problem is solved by dividing domain of web browser according to the accessing web site. The most difficult remaining problem is proof of security of the architecture; the security is discussed in Section 6.4, but it is not formalized proof, and it is desirable to give a formalized proof.

Chapter 7

Conclusions

7.1 Concluding Remarks

In this dissertation, solutions to the problems of the existing countermeasures for the threats to confidentiality and authenticity of documents are described. The solutions are not limited to improvement of security, but they also include improvement of convenience of end user.

Chapter 1 has listed the security threats to confidentiality and authenticity of documents from view of document location, and then existing countermeasures are classified into two categories, protection by document itself and protection outside document. As problems of the first category, those of management of private and public keys and protection of structured documents are pointed out. As problems of the second category, problems of access control of documents on servers on networks composed of multiple domains and convenience of an end user of a system to protect evolving attacks are also described. Finally research strategies to solve the problems have been shown.

In Chapter 2, the scheme of MSG of DSA without simultaneous users' operations has been proposed. With the prior calculation of the random part of a DSA signature, the key holders can sign a document with one-round, sequential signature operations of them. A prototype on a smartcard has shown that the performance is suitable for practical use. Security consideration against the adaptive chosen message attack and application to other signature scheme have also been discussed.

In Chapter 3, the attribute with validity period and certificate verification service with time stamp have been introduced. The trusted third party service provides the evidence that a public key certificate of a signer was valid and that a signed document existed at a certain time point, therefore, the signature of the document was valid at that point. A prototype of the service has shown that a single server has performance to server hundreds of thousands end users.

In Chapter 4, firstly the security requirements of document interchange have been defined, and the security of ODA which supports encryption and multiple signing of parts of office document was described. Next, differences of syntax between ODA and PKI, contradiction of the ODA standard, problem of integration with an existing ODA

editor have been discussed in details.

In Chapter 5, firstly requirements of a document server have been specified. Next, authorization scheme with the combination of a PAC and a CAP, and privilege delegation scheme crossing boundary of domains are proposed. Attributes in the PAC and CAP, which specify privilege of user, restriction of the privilege, and condition of object allowed to access, have been classified into six categories and access control decision has been formulated, and as a result delegation crossing security domain boundary has been realized.

In Chapter 6, Windows Vault is described, which is comprised of two Windows workstations running on virtual machines separated by secure OS. The two Windows' are connected with gateways implementing encapsulation of attached files of e-mail which may contain virus, and safe integration of e-mail clients is realized, in order to realize that the user can read all the messages on the safe workstation with ordinary operations. Performance and security of the gateway have been evaluated in details.

7.2 Future Directions

The author has come to the consideration that there are two directions toward the further study.

(1) Formal Proof of Security

Security of the proposed systems has been evaluated but they are not formally proved except the security of the MSG of DSA against the chosen message attack described in Chapter 2. The proof is of the MSG scheme, not of the implementation of the prototype on a smartcard. Generally it is very difficult to prove that a system is implemented exactly as specification, and that the system in the specification is secure as expected. In order to prove the security of the system, it is firstly required to make a model of the system, to define security on that model, and then to prove the security of the model. The author considers that it is possible to make models of the privilege delegation and authorization scheme in Chapter 5 and Windows Vault in Chapter 6, and to prove the security of the models. This is the first future study.

(2) Isolation of Web Sites

Currently the attack caused by vulnerabilities of web servers becomes a big problem; there are many servers hosting viruses which are embedded through attack to the vulnerabilities. Some vulnerability leads to leakage of secret data of end users; if a user

clicks a trap on a compromised web page, secret data on the web browser of the user, such as bank account information, is sent to attacker's site. The second future direction is an extension of data or system isolation described in Chapter 6; in order to isolate each secret, such as account information and balance of a bank, from another web site, the web servers accessed from the third workstation processing unsafe secret are separated automatically according to the secret, and then each secret is only sent to the corresponding web site.

Acknowledgements

I would like to thank Professor Norihisa Komoda of the Graduate School of Information Science and Technology at Osaka University for his countless suggestions and constructive comments on this research activity and on writing this dissertation.

I am cordially grateful to Professors Toru Fujiwara, Shinji Shimojo, Shojiro Nishio, Fumio Kishino, and Associate Professor Masanori Akiyoshi of the Graduate School of Information Science and Technology at Osaka University for their numerous suggestions for revising this dissertation.

My sincere appreciation also goes to Vice President and Executive Officer Hiroyuki Maezawa of Hitachi Software Engineering Co., Ltd. for giving full accommodation to study in the Graduate School of Information Science and Technology at Osaka University.

I would like to also express my gratitude to Professor Peter T. Kirstein of the Department of Computer Science at University College London for his primary supervision and valuable suggestions during my visit, and to Vice President and Professor Jun Murai of Keio University for giving full accommodation to study in the Department of Computer Science at University College London. I would like to tender my acknowledgments to Professor Tsutomu Matsumoto of the Graduate School of Environment and Information sciences at Yokohama National University, Professor Hiroaki Kikuchi of Department of Information Media Technology at Tokai University, Mine Sakurai of NEC Corporation for their valuable comments to my research.

I would like to express my appreciation to Dr. Atsushi Kawasaki, ex-executive director of Hitachi Software Engineering Co., Ltd., Professor Kazuko Oyanagi of Institute of Information Security, and Dr. Takashi Onoyama of Hitachi Software Engineering Co., Ltd. for their support and many useful advices ever since my joined Hitachi Software Engineering Co., Ltd. I would like to also give my thanks to my seniors, colleagues, and juniors of the company.

Finally I would like to show my deepest gratitude to my wife Yumiko, who has been encouraging and supportive; I would like to pay my heartfelt respects and gratitude to my mother Emi who died during writing this dissertation.

References

[AC2001a] C. Adams, P. Sylvester, M. Zolotarev, and R. Zuccherato, Internet X.509 Public Key Infrastructure, Data Validation and Certification Server Protocols, IETF RFC 3029, 2001.

[AC2001b] C. Adams, P. Cain, D. Pinkas, and R. Zuccherato, Internet X.509 Public Key Infrastructure, Time-Stamp Protocol (TSP), IETF RFC 3161, 2001.

[ACT2003] Act on the Protection of Personal Information (in Japanese), Law no.57, http://www5.cao.go.jp/seikatsu/kojin/houritsu/index.html, 2003.

[ACT2004] Act on Use of Information and Communications Technology in the Course of Retaining, etc. Documents Conducted by Private Entities (in Japanese), Law no.149, http://www.cas.go.jp/jp/hourei/houritu/e-bunsyo_h.html, 2004.

[AF1993] F. Anklesaria, M. McCahill, P. Lindner, D. Johnson, D. Torrey, and B. Alberti, The Internet Gopher Protocol, a Distributed Document Search and Retrieval Protocol, IETF RFC 1436, 1993.

[ANSI1988] ANSI Z39.50, 1988.

[ARGUS2001] Argus Systems Group, Inc., PitBull .comPack, OS-level Security for Solaris and AIX, White Paper, 2001.

[BBN1990] BBN, *SLATE: Multimedia Document Communication System Reference, Manual Version 1.2*, Bolt, Berank and Newman, 1990.

[BF1993] F. Borenstein, MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies, IETF RFC 1521, 1993.

[BR1997] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, Resource reservation protocol (RSVP), IETF RFC 2205, 1997.

[BS1998] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, An Architecture for Differentiated Services, IETF RFC 2475, 1998.

[BT1993] T. Berners-Lee, R. Cailliau, and N. Pellow, A Secret, The World-Wide Web Initiative, in *Proceedings of INET'93*, DBC1-5, 1993.

[CC2002] CertCo Inc., Unique Technology, http://www.certco.com/uniquetech.shtml, 2002.

[CCITT1988] CCITT, The Directory - Authentication Framework, Recommendation X.509, 1988.

[CD2002a] D. Chadwick and A. Otenko, RBAC Policies in XML for X.509 Based Privilege Management, in *Proceedings of IFIP TC11 17th International*

Conference on Information Security (SEC2002), pp.39-53, 2002.

[CD2002b] D. Chadwick, An X.509 Role-based Privilege Management Infrastructure, in *Business Briefing: Global Infosecurity 2002*, http://www.permis.org/files/article1_chadwick.pdf, 2002.

[CDW] Crystal Dew World, http://crystalmark.info/?lang=en.

[CM1993] M. Cerecedo, T. Matsumoto, and H. Imai, Efficient and Secure Multiparty Generation of Digital Signatures based on Discrete Logarithms, *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Science*, vol.E76-A, no.4, pp.532-545, 1993.

[CM2005] M. Christodorescu, S. Jha, S. Seshia, D. Song, and R. Bryant, Semantics-Aware Malware Detection, in *Proceeding of 2005 IEEE Symposium on Security and Privacy*, pp.32-46, 2005.

[CX1994] X. Cao, Realization Probabilities, the Dynamics of Queuing Systems, Springer-Verlag, 1994.

[DD1983] Department of Defense, Trusted Computer System Evaluation Criteria, CSC-STD-001-83, 1983.

[DT1999] T. Dierks and C. Allen, The TLS protocol version 1.0, IETF RFC 2246, 1999.

[DW1976] W. Diffie and M. Hellman, New Directions in Cryptography, *IEEE Transactions on Information Theory*, vol.IT-22, pp.644-654, 1976.

[ECMA1989] ECMA, Security in Open Systems - Data Elements and Service Definitions Standard, ECMA-138, 1989.

[ECMA2006] ECMA, Office Open XML Format, ECMA-376, 2006.

[FN1996] N. Freed and N. Borenstein, Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, IETF RFC 2045, 1996.

[FS1999] S. Farrell and R. Housley, An Internet Attribute Certificate Profile for Authorization, INTERNET-DRAFT, draft-ietf-pkix-ac509prof-01.txt, 1999 (work in progress).

[FS2006] S. Frantzen, Targeted Attack: Experience from the Trenches, The SANS Institute, http://isc.sans.org/diary.html?storyid=1345, 2006.

[GJ2005] J. Griffin, T. Jaeger, R. Perez, R. Sailer, L. Doorn, and R. Caceres, Trusted Virtual Domains: Toward Secure Distributed Services, The First Workshop on Hot Topics in System Dependability, 2005.

[GM1990] M. Gasser and E. McDermott, An Architecture for Practical Delegation in a Distribution System, in *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, 1990.

[GL2006] L. Gordon, M. Loeb, W. Lucyshyn, and R. Richardson, 2006 CSI/FBI

Computer Crime and Security Survey: Computer Security Institute, 2006.

[GN1999] N. Gilboa, Two party RSA key generation, in *Proceedings of Crypto'99*, pp. 116-129, 1999.

[GR1996] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, Robust Threshold DSS Signatures, in *Proceedings of Eurocrypt '96*, pp.354-371, 1996.

[GS1990] S. Golkar, et. al., The Specification of Security Facilities for Securing Whole ODA Documents, Task 2/2/6, TR111, internal document of Piloting of the Office Document Architecture, ESPRIT Project 2374, 1990.

[GS1991] S. Golkar, P. Kirstein, and A. Montaser-Kohsari, ODA Activities at University College London, *Computer Networks and ISDN Systems,* vol.21, no.3, pp.187-196, 1991.

[HK1995] K. Hickman and T. Elgamal, The SSL Protocol, Internet Draft, 1995.

[HP2004] HP NetTop: A Technical Overview, http://h20331.www2.hp.com/enterprise/downloads/HP_NetTop_Whitepaper2.pdf, 2004.

[HR1999] R. Housley, W. Ford, W. Polk, and D. Solo, Internet X.509 Public Key Infrastructure Certificate and CRL Profile, IETF RFC 2459,1999.

[HS1991] S. Haber and W. Stornetta, How to Time-Stamp a Digital Document, in *Proceedings of Crypto '90*, pp.437-455, 1991.

[IEEE1999] IEEE, Standard Specifications for Public Key Cryptography, Draft Version 13, 1999.

[IH2006] H. Inoue, A. Ikeno, M. Kondo, J. Sakai, and M. Edahiro, VIRTUS: A New Processor Virtualization Architecture for Security-Oriented Next-Generation Mobile Terminals, in *Proceedings of the 43rd annual conference on Design automation*, pp.484-489, 2006.

[ISO1988] ISO, Information Processing - Open Systems Interconnection, Text and Office Systems - Office Document Architecture (ODA), IS-8613, 1988.

[ISO1990] ISO, Revised Text of ISO 8613/DAD 4, Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Addendum 4: Security, 1990.

[ISO1991] ISO, Information Processing - Open Systems Interconnection, The Digital Signature Scheme giving Message Recovery, IS-9676, 1991.

[ISO1993] ISO, Authentication and Privilege Attribute Security Application with Related Key Distribution Functions - Part 3: Service Definitions, Working Draft, 1993.

[ISO1994] ISO, Information Processing - Open Systems Interconnection - Security Frameworks in Open Systems Part3: Access Control, Draft International Standard,

DIS-10181-3, 1994.

[ISO1995] ISO/IEC JTC 1/SC 21/WG 4 and ITU-T Q15/7, Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, Amendment 1: Certificate Extensions, Draft Amendment 1 to ITU X.509 and ISO/IEC 9594-8, 1995.

[ITU2001] ITU-T, The Directory - Authentication Framework, Recommendation X.509, 2001.

[KB1989] B. Kahle, Wide Area Information Server Concepts, Technical Report, Thinking Machines Ltd, 1989.

[KB1992] B. Kaliski, The MD2 Message-Digest Algorithm, IETF RFC 1319, 1992.

[KH1999] H. Kikuchi, K. Abe, and S. Nakanishi, Performance Evaluation of Certificate Revocation Using k-Valued Hash Tree, in *Proceedings of the Second International Workshop on Information Security (ISW'99)*, Springer, LNCS 1729, pp.103-117, 1999.

[KN1994] N. Koblitz, *A Course in Number Theory and Cryptography, Second Edition*, Springer-Verlag, 1994.

[KP1993] P. Kirstein and P. Williams, Preparing to Pilot OSI Authentication and Security Services on a Medium-scale, in *Proceedings of 4th Joint European Networking Conference,* pp.50-54 1993.

[KP1994] P. Kaijser, T. Parker, and D. Pinkas, SESAME: The Solution to Security for Open Distributed Systems, *Computer Communications*, vol.17, no.7, pp.501-518, 1994.

[KS1991a] S. Kille and J. Onions, *The PP Manual, Version 6,* University College London, 1991.

[KS1991b] S. Kille, C. Robbins, M. Roe, and A. Turland, *The ISO development environment: User's manual version 7.0, Volume 5: QUIPU*, University College London, 1991.

[KS1995] S. Kille, A String Representation of Distinguished Names, IETF RFC 1779, 1995.

[LJ1992] J. Linn, Privacy Enhancement for Internet Electronic Mail: Part I Message Encryption and Authentication Procedures , IETF RFC 1421, 1992.

[LJ1995] J. Lowry, Location-independent information object security, in *Proceedings of the Symposium on Network and Distributed System Security*, pp.54-62, 1995.

[LP2001] P. Loscocco and S. Smalley, Meeting Critical Security Objectives with Security-Enhanced Linux, in *Proceedings of the 2001 Ottawa Linux Symposium*, 2001.

[MC1990] C. McCollum, J. Messing, and L. Notargiacomo, Beyond the Pale of MAC and DAC -- Defining New Forms of Access Control, in *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, pp.190-200, 1990.

[MCA2007] McAfee, Inc., Releases New Research Suggesting Data Loss Will Lead To Next Major Corporate Collapse, Press Release, http://www.symantec.com/about/news/release/article.jsp?prid=20070319_01, 2007.

[MD1989] D. Miller, Access Control by Boolean Expression Evaluation, in *Proceedings of the Fifth Annual Computer Security Applications Conference*, pp.131-139, 1989.

[MK2001] K. Miyazaki and K. Takaragi, A Threshold Digital Signature Scheme for a Smart Card Based System, *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Science*, vol.E84-A, no.1, pp.205-213, 2001.

[MM1999a] M. Malkin, T. Wu, and D. Boneh, Experimenting with shared generation of RSA keys, in *Proceedings of the 1999 Network and Distributed System Security Symposium*,1999.

[MM1999b] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP, IETF RFC 2560, 1999.

[MM1999c] M. Malkin, T. Wu, and D. Boneh, Experimenting with Shared Generation of RSA keys, in *Proceedings of the 1999 Network and Distributed System Security Symposium*, http://www.isoc.org/isoc/conferences/ndss/99/proceedings/papers/malkin.pdf, 1999.

[MP1992] P. Marshall, WAIS: The Wide Area Information Server or Anonymous What???, 1992.

[MP2001] P. MacKenzie and M. K. Reiter, Two-Party Generation of DSA Signatures, in *Proceedings of CRYPTO 2001*, pp.137-154, 2001.

[MR2000] R. Meushaw and D. Simard, NetTop, Commercial Technology in High Assurance Applications, NSA Tech Trend Notes, vol.9, ed.4, pp.1-8, 2000.

[MS1992] S. Micali, Fair Public-Key Cryptosystems, in *Proceedings of CRYPTO'92*, pp. 113-138, 1992.

[MS2007] Microsoft Cooperation, Vulnerabilities in Microsoft Excel Could Allow Remote Code Execution (927198), Microsoft Security Bulletin, MS07-002, 2007.

[NC1993] C. Neuman, Proxy-Based Authorization and Accounting for Distributed Systems, In *Proceedings of the 13th International Conference on Distributed Computing Systems*, pp.283-291, 1993.

[NIST1977] National Bureau of Standards, Data Encryption Standard, FIPS 46, 1977.

[NIST1995] U.S. Department of Commerce/National Institute of Standards and

Technology, Secure Hash Standard, FIPS 180-1, 1995.

[NIST1998] U.S. Department of Commerce/National Institute of Standards and Technology, Digital Signature Standard, FIPS 186-1, 1998.

[NJ1991] J. Nelson, C. Bathe, I. Campbell-Grant, M. Coon, K. Fischer, P. Kirstein, G. Krönert, and M. Mabrouk, The Role of the PODA Project in the Adoption and Development of ODA, *Computer Networks and ISDN Systems,* vol.21, no.3, pp.175-185, 1991.

[OASIS2007] OASIS, Open Document Format for Office Applications (OpenDocument) v1.1, OASIS Standard, 2007.

[OSF1992] Open Software Foundation, Introduction to OSF DCE, Revision 1.0, 1992.

[PC1996] C. Park and K. Kurosawa, New ElGamal Type Threshold Digital Signature Scheme, *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Science*, vol. E79-A, no.1, pp.86-93, 1996.

[PD1993] D. Pinkas, T. Parker, and P. Kajser, Secure European System for Applications in a Multivendor Environment - an Introduction, Issue 1.2, 1993.

[PJ1982] J. Postel, Simple Mail Transfer Protocol, IETF RFC 821, 1982.

[PT1991a] T. Pedersen, Distributed Provers with Applications to Undeniable Signatures, in *Proceedings of Eurocrypt'91*, pp.221-238, 1991.

[PT1991b] T. Pedersen, A Threshold Cryptosystem without a Trusted Party, in *Proceedings of Eurocrypt'91*, pp.522-526, 1991.

[RB1999] B. Ramsdell, S/MIME Version 3 Message Specification, IETF RFC 2633, 1999.

[RR1978] R. Rivest, A. Shamir, and L. Adleman, A Method for Obtaining Digital Signatures and Public Key Cryptosystems, *Communications of ACM*, vol.21, pp.120-126, 1978.

[RR1992] R. Rivest, The MD5 Message-Digest Algorithm, IETF RFC 1319, 1992.

[RSA1993a] RSA Data Security, Inc., Public-Key Cryptography Standards #1: RSA Encryption Standard, 1993.

[RSA1993b] RSA Data Security, Inc., Public-Key Cryptography Standards #7: Cryptographic Message Syntax Standard, 1993.

[RT1990] M. Rose, *The Open Book, A Practical Perspective on OSI*, Prentice-Hall, 1990.

[SB1996] B. Schneier, *Applied Cryptography, Second Edition*, John Wiley & Sons, Inc., 1996.

[SC1989] C. Schnorr, Efficient Identification and Signatures for Smart Cards, in *Proceedings of CRYPTO '89*, pp.239-252, 1989.

[SD2004] D. Snelling, S. Berghe, and V. Li, Explicit Trust Delegation: Security for Dynamic Grids, *Fujitsu Science Technical Journal*, vol.40 no.2, pp.292-294, 2004.

[SH2001] H. Saisho, Y. Sameshima, and T. Matsumoto, Efficiency Considerations for Multiparty DSA Signature Generation System Using Smart Cards, in *Proceedings of Computer Security Symposium*, pp.349-354, 2001.

[SI1993] I. Schiller, An Alternative PEM MIME Integration, Internet Draft, 1993.

[SN1998] N. Shigechika, O. Nakamura, N. Sasagawa, and J. Murai, Construction of the Network and the Information Service System for the Nagano Olympic (in Japanese), *Journal of Information Processing Society of Japan*, vol.39 no.2 pp.86-90, 1998.

[SY1995] Y. Sameshima and P. Kirstein, Secure Document Interchange - A Secure User Agent, in *Proceedings of TERENA, 6th Joint European Networking Conference*, pp.323-1-10, 1995.

[SY1996a] Y. Sameshima and P. Kirstein, Secure Document Interchange: A Secure User Agent, *Computer Networks and ISDN Systems*, vol.28, no.4, pp.513-523, 1996.

[SY1996b] Y. Sameshima, Security Architecture based on Secret Key and Privilege Attribute Certificates, in *Proceedings of the IFIP/IEEE International Conference on Distributed Platforms*, pp.357-369, 1996.

[SY1997a] Y. Sameshima, Problems and Solution of X.509 Authentication Framework (in Japanese), in *Proceedings of the 1996 Symposium on Cryptography and Information Security*, 8B, 1997.

[SY1997b] Y. Sameshima, A Key Escrow System of the RSA Cryptosystem, in *Proceedings of the First International Information Security Workshop 1997 (ISW'97)*, pp.135-146, 1997.

[SY1997c] Y. Sameshima and P. Kirstein, Authorization with Security Attributes and Privilege Delegation: Access Control beyond the ACL, *Computer Communications*, vol.20, no.5, pp.376-384, 1997.

[SY1997d] Y. Sameshima and H. Miyazaki, Privacy Enhanced Message System using Secret-Key and User-Attribute Certificates (in Japanese), in *Proceedings of the 5th Workshop on Multimedia and Distributed Systems*, pp.85-92, 1997.

[SY2000] Y. Sameshima and T. Tsutsumi, Reducing Certificate Revocation and Non-Repudiation Service in Public Key Infrastructure, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol.E83-A, no.7, pp.1441-1449, 2000.

[SY2001] Y. Sameshima, H. Saisho, and T. Matsumoto, Multiparty DSA Signature Generation without Concurrent Processing (in Japanese), *Technical Report of IEICE*, ISEC 2001-66, pp.97-104, 2001.

[SY2004] Y. Sameshima, H. Saisho, K. Oyanagi, and T. Matsumoto, Multiparty DSA Signature Generation without Simultaneous User Operations, *IEICE Transaction on*

Information and Systems, vol. E87-D, no.8, pp.2095-2105, 2004.

[SY2005] Y. Sameshima and H. Saisho, Prevention of Information Leakage and Virus Infection with use of Secure OS and Virtual Machine (in Japanese), in *Proceedings of the 13th Workshop on Multimedia and Distributed Systems*, pp.151-155, 2005.

[SY2007] Y. Sameshima, H. Saisho, T. Matsumoto, and N. Komoda, Windows Vault: Prevention of Virus Infection and Secret Leakage with Secure OS and Virtual Machine, in *Pre-Proceedings of the 8th International Workshop of Information Security Applications 2007 (WISA 2007)*, pp.249-261, 2007.

[SYM2007] Symantec Reports Rise in Data Theft, Data Leakage, and Targeted Attacks Leading to Hackers' Financial Gain, News Release, http://www.symantec.com/about/news/release/article.jsp?prid=20070319_01, 2007.

[TCG2007] Trusted Computing Group, TCG Specification Architecture Overview, Specification Revision 1.4, 2007.

[VP1996] P. Vixie, Name server operations guide for BIND, release 4.9.5, Internet Software Consortium, 1996.

[WP1994] P. Williams, et. al., *The OSI Security Package: OSISEC User's Manual, Release 2.3*, University College London, 1994.

[YH1995] H. Yu and M. Burati, *DCE 1.2 Global Groups Functional Specification*, Open Software Foundation, RFC 87.0, 1995.

[YY1996] Y. Yamane and K. Sakurai, How to restrict investigators' tapping in Key Escrow Systems (in Japanese), in *Proceedings of the 1996 Symposium on Cryptography and Information Security*, 7C, 1996.

Appendix

A. Verifiable Secret Sharing

In this section, the $(t, t+1, M)$ verifiable secret sharing [PT1991a, PT1991b] is described shortly. It is assumed $M \geq t+1$.

A.1 Distribution of Secret Shares

Firstly the dealer distributes the secret $s \in Z_q$ as follows:

(1) chooses a random polynomial f of degree t over Z_q:

$$f(x) = f_0 + f_1 x + \cdots + f_t x^t$$

where, $f_0 = s$, $f_1, f_2, \cdots, f_t \in Z_q$, and $f_t \neq 0$,

(2) computes the secret shares $s_i = f(i) \in Z_q \ (1 \leq i \leq M)$ and the public shares $F_m = \exp(G, f_m) \in Z_q \ (0 \leq m \leq t)$, and

(3) sends s_i and $F_0, \cdots, F_t$ to U_i in secure form.

A.2 Verification of Distributed Secret Shares

Secondly each key holder U_i verifies his secret share s_i:

(1) U_i verifies that

$$\exp(G, s_i) = (\prod_{m=0}^{t} \exp(F_m, i^m)) \bmod p.$$

(2) If the verification fails, then the secret sharing stops. Otherwise, U_i broadcasts $F_0, \cdots, F_t$ to the other key holders.

(3) U_i confirms that the received $F_0, \cdots, F_t$ were sent from the other key holders, and that they are the same as the ones U_i received from the dealer. If the confirmation fails, then the secret sharing stops.

A.3 Secret re-Construction

Since there is one and only one polynomial of degree t satisfying $f(i) = s_i \bmod q$ for $t+1$ values of i, any $t+1$ key holders, for example $U_1, \cdots, U_{t+1}$, can reconstruct the secret with the Lagrange formula, where $j \in \{1, \cdots, t+1\}$:

$$\begin{aligned} f(x) &= \sum_{i=1}^{t+1} \prod_{j \neq i} \frac{j-x}{j-i} f(i) \bmod q \\ &= \sum_{i=1}^{t+1} \prod_{j \neq i} \frac{j-x}{j-i} s_i \bmod q, \end{aligned}$$

and then

$$s \equiv f(0) \equiv \sum_{i=1}^{t+1} \prod_{j \neq i} \frac{j}{j-i} s_i \pmod q.$$

B. Proof of Theorem 1

Put $b^{(l)}(x)$, $c^{(l)}(x)$, $v^{(l)}(x)$, and $h(x)$ as follows:

$$b^{(l)}(x) = \sum_{j=1}^{n} b_j^{(l)}(x) \bmod q,$$

$$c^{(l)}(x) = \sum_{j=1}^{n} c_j^{(l)}(x) \bmod q,$$

$$v^{(l)}(x) = \sum_{j=1}^{n} v_j^{(l)}(x) \bmod q, \text{ and}$$

$$h(x) = \sum_{j=1}^{n} h_j(x) \bmod q.$$

Then

$$d_i^{(l)} = ((b^{(l)}(i))(c^{(l)}(i)) + v^{(l)}(i)) \bmod q,$$

$$\begin{aligned} r_i^{(l)} &= \exp(\exp(G, \sum_{j=1}^{n} b_j^{(l)}(0)), (b^{(l)}(0))^{-1}(c^{(l)}(0))^{-1} \bmod q) \bmod q \\ &= \exp(\exp(G, b^{(l)}(0)), (b^{(l)}(0))^{-1}(c^{(l)}(0))^{-1} \bmod q) \bmod q \\ &= \exp(G, (c^{(l)}(0))^{-1} \bmod q) \bmod q \\ &= r^{(l)}, \end{aligned}$$

$$\begin{aligned} s^{(l)} &= ((hash(mssg^{(l)}) + r^{(l)} h(0))\, c^{(l)}(0)) \bmod q \\ &= ((hash(mssg^{(l)}) + r^{(l)} \sum_{i=1}^{n} h_{i,0})\, c^{(l)}(0)) \bmod q. \end{aligned}$$

Comparing the original DSA, it is clear that $(r^{(l)}, s^{(l)})$ is a valid signature.

C. View of the MSG

The following is the view of U_1. The suffix (l) is omitted.

(1) The view of U_1 during the key generation procedure is as follows:

- h_1 (polynomial of degree t)
- $a_{1,j} = h_1(j) \bmod q \ (1 \le j \le n)$
- $A_j \ (1 \le j \le n)$
- $A_{j,m} (1 \le j \le n \ , 0 \le m \le t)$
- $K_{1,j} (2 \le j \le n)$
- $\{a_{1,j}\}_{E(1,j)} (2 \le j \le n)$
- $\{a_{j,1}\}_{E(j,1)}$ and $a_{j,1} \ (2 \le j \le n)$
- $\{A_1'\}_{S(1,j)}$ and $\{A_j'\}_{S(j,1)} (2 \le j \le n)$

(2) The view of U_1 during the random sharing procedure is as follows:

- b_1, c_1, v_1, and w_1
- $b_{1,j} = b_1(j) \bmod q \ (1 \le j \le n)$
- $c_{1,j} = c_1(j) \bmod q \ (1 \le j \le n)$
- $v_{1,j} = v_1(j) \bmod q \ (1 \le j \le n)$
- $w_{1,j} = w_1(j) \bmod q \ (1 \le j \le n)$
- $B_{j,m}$ and $C_{j,m} (1 \le j \le n, 0 \le m \le t)$
- $V_{j,m}$ and $W_{j,m} (1 \le j \le n, 1 \le m \le 2t)$

- CMT_1 and $\{CMT_1\}_{S(i,j)}\ (2 \le j \le n)$
- $\{CMT_j\}_{S(j,1)}\ (2 \le j \le n)$
- $\{b_{1,j}, c_{1,j}, v_{1,j}, w_{1,j}\}_{E(1,j)}$ and $\{b_{j,1}, c_{j,1}, v_{j,1}, w_{j,1}\}_{E(j,1)} (2 \le j \le n)$
- $b_{j,1}$, $c_{j,1}$, $v_{j,1}$, and $w_{j,1} (2 \le j \le n)$
- $\{BCVW_1\}_{S(1,j)}$ and $\{BCVW_j\}_{S(j,1)} (2 \le j \le n)$
- d_1

(3) The view of U_1 during the signature generation procedure is as follows:

- $mssg$
- $d_j\ (2 \le j \le n)$
- r_1
- $hash(mssg)$
- s_1

D. Proof of Theorem 2

Without loss of generality, it is assumed that Y corrupts $U_1, \cdots, U_t$. X is constructed which uses Y as a subroutine. X provides Y with (p, q, G, P) and the content of the random tape of Y. X generates the following view of the key generation procedure, where the output public key of the n key holders is P, and provides it to Y:

(1) generates t polynomials of degree t $h_1, \cdots, h_t$ as Y does,

(2) calculates $A_{1,0} = \exp(G, h_1(0)), \cdots, A_{t,0} = \exp(G, h_t(0))$,

(3) generates randomly $A_{t+1,0}, \cdots, A_{n-1,0} \in Z_p$ with some $h_{t+1}(0) \in Z_p$ such that $A_{t+1,0} = \exp(G, h_{t+1}(0))$, etc., and calculates $A_{n,0} \in Z_p$ satisfying

$$P = (\prod_{j=1}^{n} A_{j,0}) \bmod p,$$

(4) calculates $K_{i,j}\ (1 \le i \le t,\ 1 \le j \le n, i \ne j)$,

(5) calculates the following:

- $a_{i,j} = h_i(j) \bmod q\ (1 \le i \le t, 1 \le j \le n)$,
- $\{a_{i,j}\}_{E(i,j)} (1 \le i \le t, 1 \le j \le n)$,
- $A_{i,m} = (1 \le i \le t, 0 \le m \le t)$,

(6) generates randomly $a_{j,i} \in Z_q (1 \le i \le t < j \le n)$,

(7) calculates $\{a_{j,i}\}_{E(j,i)} (1 \le i \le t < j \le n)$,

(8) calculates $A_{j,m} = (t < j \le n, 0 \le m \le t)$, satisfying the following:

$$\exp(G, a_{j,i}) = (\prod_{m=0}^{t} \exp(A_{j,m}, i^m)) \bmod p$$

with $1 \le i \le t$, and

(9) calculates the following:

- $A_j = hash(A_{j,0}, \cdots, A_{j,t})\ (1 \le j \le n)$,
- $A_j' = hash(A_{1,0}, \cdots, A_{n,t})\ (1 \le j \le n)$,
- $\{A_i\}_{S(i,j)}\ (1 \le i \le t, 1 \le j \le n, i \ne j)$,
- $\{A_j\}_{S(j,i)}\ (1 \le i \le t, 1 \le j \le n, i \ne j)$,
- $\{A_i'\}_{S(i,j)}\ (1 \le i \le t, 1 \le j \le n, i \ne j)$, and
- $\{A_j'\}_{S(j,i)}\ (1 \le i \le t, 1 \le j \le n, i \ne j)$.

Note that the probability distribution of this view is identical to the one of the adversary. This is because the polynomials of degree t are generated as Y does in

Step (1), and because $A_{j,0}$ in Step (3) and $a_{j,i}$ in Step (6) $(1 \le i \le t < j \le n)$ are generated randomly whose probability distribution is as the same as the one generated by the honest key holders $U_j (t \le j \le n)$.

Next X calls Y as a subroutine and gives the view generating the signatures (r', s')'s of the chosen messages $mssg$'s. X calculates the first parameters of the $(l+1)$ through $(l+4)$-th signatures with the knowledge of r's got from the oracle before the actual signatures (r', s')'s are given during the signature generation procedure. Because each of views of the signature generations is independent from the other signature generation, the l-th signature is focused to and the suffix (l) is omitted. If Y requests a signature of a message $mssg'$, X obtains the signature (r', s') from the oracle. Note that Y knows the first parameters and chooses messages before the oracle gives the signatures. X generates the following view of the parameter sharing and signature generation procedures where the output signature is (r', s') and provides it to Y:

(1) generates polynomials of degree t as Y does:

- $b_i (1 \le i \le t)$,
- $c_i (1 \le i \le t)$,

(2) calculates the following:

- $b_{i,j} = b_i(j) \bmod q \ (1 \le i \le t, 1 \le j \le n)$,
- $c_{i,j} = c_i(j) \bmod q \ (1 \le i \le t, 1 \le j \le n)$,
- $B_{i,m} = \exp(G, the\ m-th\ coefficient\ of\ b_i) \ (1 \le i \le t, 0 \le m \le t)$,
- $C_{i,m} = \exp(G, the\ m-th\ coefficient\ of\ c_i) \ (1 \le i \le t, 0 \le m \le t)$,

(3) generates polynomials v_i and w_i of degree $2t$ whose constant terms are zero as Y does, and calculates the following:

- $v_{i,j} = v_i(j) \bmod q \ (1 \le i \le t, 1 \le j \le n)$,
- $w_{i,j} = w_i(j) \bmod q \ (1 \le i \le t, 1 \le j \le n)$,
- $V_{i,m} = \exp(G, the\ m-th\ coefficient\ of\ v_i) \ (1 \le i \le t, 1 \le m \le 2t)$,

- $W_{i,m} = \exp(G, the\ m-th\ coefficient\ of\ w_i)\ (1 \le i \le t, 1 \le m \le 2t)$,

(1) calculates

- $\{b_{i,j}, c_{i,j}, v_{i,j}, w_{i,j}\}_{E(i,j)} (1 \le i \le t, 1 \le j \le n)$

(2) generates randomly $b_{j.i}, c_{j,i}, v_{j,i}, w_{j,i}\ (1 \le i \le t < j \le n)$, and calculates

- $\{b_{j,i}, c_{j.i}, v_{j,i}, w_{j,i}\}_{E(j,i)} (1 \le i \le t < j \le n)$,
- $d_i = ((\sum_{j=1}^{n} b_{j,i})(\sum_{j=1}^{n} c_{j,i}) + \sum_{j=1}^{n} v_{j,i}) \bmod q\ (1 \le i \le t)$,

(3) generates randomly a polynomial d of degree $2t$ satisfying $d(i) \equiv d_i \pmod q\ (1 \le i \le t)$, and then calculates $d_j \equiv d\ (j) \pmod q\ (t \le j \le n)$,

(4) calculates $r'^* = (\exp(G, hash(mssg')s'^{-1})) \bmod p$,

(5) generates randomly $B_{j,0} (t < j \le n)$ satisfying

$$\begin{aligned} r'^* &= \exp(\prod_{j=1}^{n} B_{j,0}, (\sum_{j=1}^{n} \prod_{k \ne j} \frac{k}{k-j} d_j)^{-1} \bmod q) \\ &= \exp(\prod_{j=1}^{n} B_{j,0}, d(0)^{-1} \bmod q) \end{aligned}$$

in the same way as the case of $A_{j,0}$,

(6) calculates $B_{j,m} (t < j \le n, 1 \le m \le t)$ satisfying

$$\exp(G, b_{j,i}) = (\prod_{m=0}^{t} \exp(B_{j,m}, i^m)) \bmod p$$

with $1 \le i \le t$,

(7) calculates $C_{j,m}, V_{j,m}, W_{j,m} (t < j \le n)$ satisfying

$$\exp(G, c_{j,i}) = (\prod_{m=0}^{t} \exp(C_{j,m}, i^m)) \bmod p,$$

$$\exp(G, v_{j,i}) = (\prod_{m=1}^{2t} \exp(V_{j,m}, i^m)) \bmod p,$$

$$\exp(G, w_{j,i}) = (\prod_{m=1}^{2t} \exp(W_{j,m}, i^m)) \bmod p,$$

with $1 \le i \le t$,

(8) calculates the following:

$\{CMT_i\}_{S(i,j)}$,

$\{CMT_j\}_{S(j,i)}$,

$\{BCVW\}_{S(i,j)}$, and

$\{BCVW\}_{S(i,j)} (1 \le i \le t < j \le n, i \ne j)$,

(9) calculates and $s_i (1 \le i \le t)$, and

(10) generates a polynomial s of degree $2t$ satisfying $s(i) = s_i \bmod q \ (1 \le i \le t)$ and

$s(0) = s'$ and then puts $s_j = s(j) \bmod q \ (t \le j \le n)$.

Note that the probability distribution of this view is identical to the one of the adversary Y, because of the same reason in the case of the key generation. Now X obtains the whole view and gives it to Y, which outputs (r', s'). In this way X does the adaptive chosen message attack, and it is clear that the equation (1) of Theorem 2 holds.

E. ASN.1 Definition of CVSTS Request and Response

```
CVSTSRequest ::= OPTIONALLY SIGNED SET {
    version                     [0] INTEGER DEFAULT 0,
    certificates                SEQUENCE OF SEQUENCE {
        issuer                  Name,
        serialNumber            INTEGER },
    verificationRequestTime     [1] UTCTime OPTIONAL,
    dataToBeTimeStamped         [2] OCTET STRING OPTIONAL,
    messageDigestAlgorithm      [3] AlgorithmIdentifier OPTIONAL,
    additionalInformation
                    [4] SEQUENCE OF AdditionalInformation OPTIONAL,
    requestOriginator           [5] Name OPTIONAL,
    requestIdentifier           [6] OCTET STRING OPTIONAL }
```

```
CVSTSResponse ::= SEQUENCE { SIGNED SET {
    version                     [0] INTEGER DEFAULT 0,
    certificates                SEQUENCE OF SEQUENCE {
        issuer                  Name,
        serialNumber            INTEGER,
        lastUpdate              UTCTime },
    requestIdentifier           [1] OCTET STRING OPTIONAL,
    verificationResult          VerificationResult,
    invalidCertificate          [2] SEQUENCE {
            issuer              Name,
            serialNumber        INTEGER } OPTIONAL,
    revokedReason               [3] RevokedReason OPTIONAL,
    revokedOrHoldTime           [4] UTCTime OPTIONAL,
    invalidTime                 [5] UTCTime OPTIONAL,
    verificationTime            UTCTime,
    dataToBeTimeStamped         [6] OCTET STRING OPTIONAL,
    messageDigestAlgorithm      [7] AlgorithmIdentifier OPTIONAL,
    additionalInformation
                    [8] SEQUENCE OF AdditionalInformation OPTIONAL,
    requestedTime               [9] UTCTime OPTIONAL,
    generationTime              UTCTime,
    responseIdentifier          [10] OCTET STRING OPTIONAL },
    serverCertificate           CertificationPath OPTIONAL }
```

F. UCL Activities in ODA and Security

F.1 PASSWORD Project

University College London (UCL) has been active in both the ODA and the security area. In the context of the Piloting ODA (PODA) series of projects [NJ1991], culminating in the PODA-SAX Project, UCL has extended an existing compound document editor, Slate [BBN1990] by adding an ODIF back-end. It has also integrated the Slate User Agent (UA) [GS1991] with several message systems, such as X.400 [KS1991] and MIME [BF1993], and with a document store, an Autonomous Active Mailbox (AAM) which stores or returns an ODA document via the message systems [GS1991]). In the context first of the PODA-SAX and later the PASSWORD Project [KP1993], UCL has also

developed an OSI security toolkit OSISEC [WP1994] and various secured applications. OSISEC is based on the X.509 Security Framework [CCITT1988]; it contains encryption libraries and tools for managers operating Certification Authorities (CAs). The applications based on OSISEC have been implemented and used in the project.

In the project, three kinds of CAs have been established; a single top level CA, policy CAs certified by the top level CA, organisational CAs certified by policy CAs. Organisational CAs issue certificates for users. All applications are required to attach a complete forward certification path to the signed content; certificates of the signer, the organisational CA which issued the signer's certificate and the policy CA which issued the certificate of the organisational CA. The recipient who has the trusted public key of the root CA can verify the signature.

Two ODA utilities have been developed during the PASSWORD Project. The original DOCSEC [GS1990] secures a document as a whole, while PDOCSEC secures each part of a document individually. Both utilities are integrated with the Slate UA. While reasonably novel when it was developed in 1990, DOCSEC contains little functionality additional to what can now be achieved with PEM/MIME [LJ1992], except the way it uses Directories to obtain certificates.

The OSI security toolkit, OSISEC [WP1994] contains libraries which implement asymmetric encryption such as RSA encryption [RSA1993a], DSS [ISO1991], the DES symmetric encryption algorithm [NIS1977], message digest algorithms MD2/5 [KB1992, RR1992], and tools for managers operating CAs. Applications based on OSISEC have been implemented and used in the PASSWORD Project. They are secured directory services (DISH/DE DUA and QUIPU DSA [KS1999b], PEM [LJ1992] and secured X.400 UA [KP1993].

F.2 Implementation of Document UA

F.2.1 Slate Multimedia UA

The Slate editor [BBN1990] from Bolt Bonarek and Newman can handle multimedia information such as text, raster-graphics, geometric-graphics, audio and spreadsheet; it is configurable to work with an external message handling system. UCL has developed Slate/ODA converters which supports the FOD026 Document Application Profile (DAP) and integrated the editor with X.400 and SMTP message handling systems [GS1991]. In these neither audio nor spreadsheet have been supported because they have not yet been included in the ODA standard.

secured document editor in order to support the security services. The other is to implement a filter which encodes an ordinary plain ODIF stream to a secured ODIF stream and decodes the secured ODIF stream to the plain ODIF stream; the filter must be integrated with the non-secure document editor.

The second choice is adopted for two reasons; first a document editor is normally very complex and hard to change; secondly, since many document editors support ODIF input and output or have filters which transform the original format into the ODIF and vice versa, the filter procedure is applicable to all such document editors.

The UCL implementation of the filter is called PDOCSEC; it consists of two programs, namely an encoder and a decoder. The encoder requires information about identities of parts of a document to be sealed and enciphered, the names of the privileged recipients, the sealing location, the date and the time. It enciphers and generates seals of specified parts of the document and produces a secured ODIF stream. The decoder verifies and deciphers parts of the document of which the associated privileged recipient is the user and produces a plain ODIF stream. The decoder also outputs information on which parts of the document are passed or failed during verification and decipherment.

F.2.3 Integration of PDOCSEC with Slate UA

To support security services on parts of a document, the Slate/ODA converters are changed to support handling some security information. For sending a secured document, a user can attach a tag to a part of the document on the screen by selecting a menu; the tag with a security service type and the names of privileged recipients is displayed on the screen. The editor stores the tag into the output Slate file as well as the document content. The security information is placed before or near a Slate object, such as text, graphics, etc., in the Slate file. The Slate-ODA filter converts the Slate file into a plain ODIF stream and produces a file including lines each of which contains security information that consists of a service type, privileged recipients and an object or class identifier (OCID). The PDOCSEC encoder reads this file and the plain ODIF stream and produces a secured ODIF stream which is sent to a message transfer system.

On receipt of a secured document, first the secured ODIF stream is converted

into a plain ODIF stream with the PDOCSEC decoder. The PDOCSEC decoder writes security information including records each of which consists of an OCID, a security service type, the names of privileged recipients, a verify and decipher result and optionally originators to an external file. The ODA-Slate filter reads this file and the plain ODIF stream, and converts into a Slate file in which security information is placed before or near a Slate object in the tag form. Note that the slate editor does not know OCIDs at all.

A secure ODIF stream is just an octet string and it can be transferred by any message systems such as X.400, MIME+SMTP. But care must be paid to handling of the whole document. Because deletion of seal information (Sealed Document Profiles, Sealed Document Bodyparts and Sealed Attributes) is undetectable, some process must be applied to the exported data stream.

There are three options to send a secured ODIF stream; DOCSEC+X.400, PKCS#7+X.400 and MIME+PEM+SMTP. Both DOCSEC and PKCS#7 support confidentiality and integrity of a whole ODA document and the secured whole document can be sent by X.400 as a bilaterally or externally defined body part. MIME [BF1993] supports ODA application subtype where ODIF is encoded into printable characters according to the base64 encoding rule. A MIME message containing ODA can be enclosed in PEM body according to the PEM-MIME integration working draft [SI1993]. Currently the third method MIME+PEM+SMTP has been implemented. Other methods will be realized later.