A Study on Read-Write Protection of a Digital Document by Cryptographic Techniques

Yasuo Hatano[1,2], Kunihiko Miyazaki[1] and Toshinobu Kaneko[2]

[1]*Hitachi, Ltd., Systems Development Laboratory.*
292, Yoshida-cho, Totsuka-ku, Yokohama-shi, Kanagawa-ken, 244-0817, Japan.

[2]*Faculty of Science and Technology, Tokyo University of Science.*
2641, Yamazaki, Noda-shi, Chiba-ken, 278-8510, Japan.

Abstract

Sensitive information, e.g., privacy information or company secret, should be carefully managed and it is desired that only privileged users can read and edit these kinds of information. For these needs, this paper proposes cryptographic schemes for a proper use of digital document and a concrete construction for the proposed schemes. Note that the proposed schemes are called "Content Protection Schemes(CPSs)" in this paper. The proposed schemes enable a sender to decide an access control, i.e., read-write protection, on a document by public keys of recipients and a recipient can read or edit a part of the document according to the access control without any help of neither the sender nor any trusted entity. The concrete construction proposed in this paper consists of only standard cryptographic techniques. This means that the proposed construction can be easily implemented by a standard cryptographic library and that users do not need to prepare a special key set for the proposed schemes. Since the proposed schemes control the read-write protection on a digital document by private keys of recipients, they can promote proper use of digital document even if we could not manage them on a server.

1. Introduction

Digital documents have been widely used for last a few decades and often contains sensitive information, e.g., personal information and national/company secret. Hence we have to manage them carefully by controlling users who can access the documents. For instance, a security rule of Health Insurance Portability and Accountability Act(HIPAA) in the U.S., which was published in 2003, requires that the confidentiality, integrity and availability of electronic protected health information, e.g., privacy information of patients, is carefully managed. In addition, since the effectuation of privacy information law in Japan, 2005, lots of guidelines have been published and established that privacy information is properly controlled when it is taken, used, updated and disclosed. Due to these laws and guidelines, we are required that privacy information is encrypted for the confidentiality or attached with a digital signature for the integrity.

Encrypting privacy information enables us to restrict users who can access privacy information and therefore achieves the confidentiality of privacy information. A technique called "eXtended Multi-Recipient Encryption Schemes(XMRESs)"[9] is a kind of partial encryptions and combines a hybrid construction and a partial encryption. XMRESs achieve whether each block composed of a document is disclosed to each of several recipients or not and can control rights to read a part of a document for recipients.

On the other hand, digital signature schemes achieve the data integrity because any alteration on a digitally signed document can be detected. However, the property of standard digital signature schemes sometime make a problem if a signed document is updated or disclosed. Several works, such as sanitizable signature schemes[1], [7], [11], [12], [18], [19], [21], address this problem[1]. In the sanitizable signature schemes, even if we should delete a part of a signed document for some requirements, e.g., privacy protection in freedom of information system, the signed document which is partially deleted is still validated. A. Saito, el. al., expanded the sanitizable signature schemes to the ones in which a user can edit a block in a signed document if the signer accepts it[18]. G. Ateniese, el. al., also improved sanitizable signature schemes[1]. The improved scheme[1] allows a signer to designate a user when he/she sign a document and the designated user can edit a part of the signed document without any help of neither the signer nor a trusted entity. These improved schemes[1], [18] enable to control rights for editing a part of a document.

Other related works are found in techniqes for access control of XML applications although they have no proof based on complexity theory such as techniques we introduced in the above. For instance, M. A. Rahanman, et. al., proposed a technique for an access control for an XML document by using cryptographic techniques[20]. In this technique, a

[1] Some of these works are called "redactable signature schemes"[12] or "content extraction signatures"[19]. In this paper, however, we call them as "sanitizable signatures" according to the several references, e.g., [21].

978-1-4244-5113-5/09/$25.00 ©2009 IEEE

document owner can control right for reading a document by encrypting a document and an authorized user can edit an XML document by collaborating the document owner. After the authorized user edits the document, the integrity of the document can be checked by the signature of the authorized user. However, the technique does not include a mechanism to control rights for editing a part of a document by a sender.

To the best of our knowledge, there are no cryptographic schemes that enable us to control both reading and writing a part of a document without any help of neither the sender nor any trusted entity. In addition, in the paper[1], although the authors show several applications for sanitizable signatures, e.g., outsourcing database, medical information, software distribution and so on, these need to control not only to edit a part of a document but also to read a part of the document.

In this paper, we propose cryptographic schemes for controlling rights not only to edit a part of a document but also to read a part of the document by using a public keys and shows a concrete construction. Note that, in this paper, we call the proposed schemes "Contents Protection Schemes(CPS)". Most of the related works, which we introduced in the above, uses sophisticated cryptographic techniques, e.g., Chameleon hash[1] or Bilinear map[7], [21]. On the other hand, the proposed construction consists of only standard cryptographic techniques: a standard asymmetric-key cryptography, symmetric-key cryptography and digital signature. This means that the proposed construction can be easily implemented by a standard cryptographic library without any modification of the cryptographic functions in the library. In addition, we do not need to prepare a special key set for an application of the proposed schemes. Since the proposed schemes enable us to control the read-write protection on a digital document without any help of a trusted entity, the proposed schemes can promote proper use of digital documents even if we could not manage them on a server.

2. Content Protection Schemes

2.1. Framework of the schemes

The proposed schemes, which we call "Content Protection Schemes" in this paper, enable us to control rights for both reading and writing a part of a document by cryptographic techniques. A framework of the proposed schemes is described in Figure 1. As Figure 1 shows, there are three entities for the proposed schemes: key center, sender and recipient.

Key center generates a signing and validation key pair for a sender and also generate encryption and decryption key pairs for recipients. A sender divides a document into several blocks and, for each of several recipients, he/she decides an access control list for read-write protection of each block. Then the sender encrypts the document according to the access control list and send the resulting ciphertext to recipients. When a recipient receives the ciphertext, he/she decrypts it with his/her decryption key and reads the blocks of the document that the sender allows. Also, in the proposed schemes, the recipient can edit some blocks that the sender allows.

CPSs consist of the following three algorithms.

Key Generation Algorithm $\texttt{KeyGen}(\lambda)$**:**
Probabilistic algorithm which accepts a security parameter λ and which outputs a signing/validation key pair of a sender (sk, vk) and encryption/decryption key pairs for recipients, $(ek_1, dk_1), \dots, (ek_u, dk_u)$.

Encryption Algorithm $\texttt{Enc}_{sk,EK}(AC, M)$**:**
Probabilistic algorithm which accepts an access control list AC, a sender's signing key sk, encryption keys $EK = (ek_1, \dots, ek_u)$ and a plaintext $M = (m_1, \dots, m_n)$, and which output a ciphertext $C = (\texttt{aux}, c_1, \dots, c_n)$, where aux denotes an auxiliary information such as AC.

Decryption Algorithm $\texttt{Dec}_{vk,dk_i}(C)$**:**
Deterministic algorithm which accepts a sender's validation key vk, a decryption key dk_i and a ciphertext C, and which outputs the decrypted plaintext M^* or decryption failed $\perp$(invalid ciphertext).

Update Algorithm $\texttt{Upd}_{vk,dk_i}(C, U)$**:**
Probabilistic algorithm which accepts a sender's validation key vk, a decryption key dk_i, a ciphertext C and update information $U = (j, m_j^+)$, and which outputs an update ciphertext C^+ or update failed $\perp$(invalid ciphertext or invalid update information).

In the following, we use the notation ES.Alg to denote the algorithm Alg of ES. For instance, CPS.Enc denotes the encryption algorithm Enc of a content protection scheme CPS.

2.2. Security requirement

In the proposed schemes, a sender decides an access control list for a document, which describes which parts of a document can be read or edited by a recipient, and chooses encryption keys of the recipients to encrypt the document according to the access control list. When a recipient decrypts a ciphertext with decryption key, he/she can read or edit blocks that the sender accepts to read or edit for the recipient. The control for read-write protection in the proposed schemes is achieved by the decryption key that the recipient has. We consider the following properties for the security of the proposed schemes.

Confidentiality:
It is infeasible to know any information about a

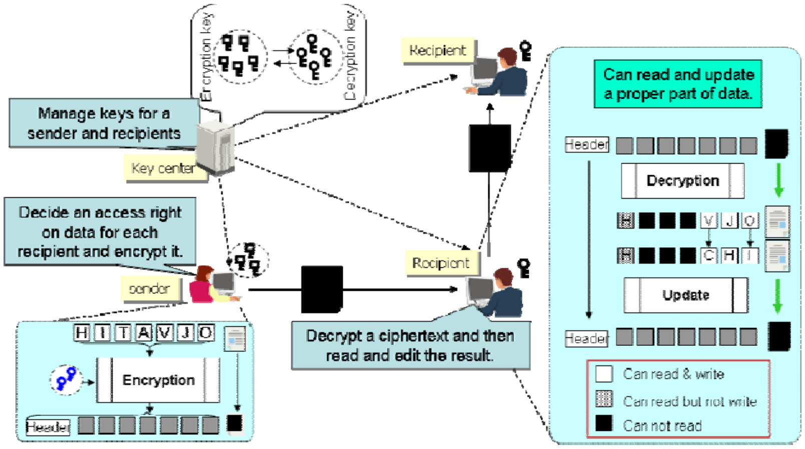
Encryption key
Decryption key
Recipient
Manage keys for a sender and recipients
Key center
Can read and update a proper part of data.
Header
Decryption
V J O
C H I
Update
Header
Can read & write
Can read but not write
Can not read
Decide an access right on data for each recipient and encrypt it.
Recipient
sender
H I T A V J O
Encryption
Header
Decrypt a ciphertext and then read and edit the result.

Figure 1. Framework of Content Protection Schemes(CPSs)

block m_i without the knowledge of decryption keys to decrypt the block m_i.

Integrity:

It is infeasible to alter a block m_i of a document and an access control list that the sender decides without the knowledge of neither decryption keys to update m_i or signing key of the sender sk.

Formal definitions of these requirements are as follows.

Definition 1(Confidentiality): Let $\mathtt{CPS}$ be a content protection scheme and an adversary $\mathcal{A}$ be a probabilistic polynomial-time Turing machine. In the first phase, given a security parameter, a validation key vk, encryption keys $EK = (ek_1, \ldots, ek_u)$ and a subset of decryption keys $DK' = (dk_1, \ldots, dk_w)$ $(w < u)$, $\mathcal{A}$ choose an access control list AC, a pair of messages M_0 and M_1. Note that the pair M_0 and M_1 only differs in a block m_i, where m_i can not be decrypted with any decryption keys in DK'. Note that both M_0 and M_1 consist of n blocks. Then, in the second phase, the adversary $\mathcal{A}$ receives a ciphertext $C_b \leftarrow \mathtt{CPS.Enc}_{EK}(AC, M_b)$, where $b(= \{0, 1\})$ is randomly chosen, and the adversary $\mathcal{A}$ guesses for b. Note that, in the first and second phase, the adversary $\mathcal{A}$ has oracle access to the encryption and the decryption of $\mathtt{CPS}$ and the adversary never makes a query for C_b to the decryption oracle. We say that the $\mathtt{CPS}$ has confidentiality if, for any adversary $\mathcal{A}$, the advantage of the adversary $\mathcal{A}$ for guessing b is negligible.

Definition 2(Integrity): Let $\mathtt{CPS}$ be a content protection scheme and an adversary $\mathcal{A}$ be a probabilistic polynomial-time Turing machine. We assume that $\mathcal{A}$ has oracle access to the encryption, the decryption and the update of $\mathtt{CPS}$. The adversary $\mathcal{A}$ is given a security parameter, a validation key vk, and encryption keys $EK = (ek_1, \ldots, ek_u)$. Then, the adversary $\mathcal{A}$ outputs a candidate of valid ciphertexts C for a message M and an access control list AC. The adversary succeeds this game if the ciphertext C is a valid ciphertext and if the ciphertext C never includes answers from the encryption oracle $\mathcal{O}^{\mathtt{Enc}}$ and c_i in C never includes answers from the update oracle $\mathcal{O}^{\mathtt{Upd}}$ during the game. We say that the $\mathtt{CPS}$ has integrity if, for any adversary $\mathcal{A}$, the success probability of the adversary $\mathcal{A}$ to win the above game is negligible.

Note that, the notion of the integrity is essentially the same as the "unforgeability" notion defined in the sanitizable signature schemes[1].

The sanitizable signature schemes also require that a security requirement called "indistinguishability", which means that, after a block of a message is updated, it is infeasible to know any original information about the block. In CPSs, this security requirement is optionally required, especially if the CPSs are used as an application of a sanitizable signature scheme. For instance, this security requirement must be required, if a recipient needs to edit a part of a ciphertext for protecting privacy information before he/she sends the ciphertext to another recipient. However, this security requirement is not always needed and therefore the description of the security requirement is omitted in this paper.

3. Concrete Construction

3.1. Components

In this section, we show a concrete construction for the proposed schemes, which consists of standard cryptographic techniques. For description of the proposed construction, we use notations $\mathtt{SE} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$ for a symmetric-key cryptography, $\mathtt{PE} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$ for an asymmetric-key cryptography and $\mathtt{DS} = (\mathtt{KeyGen}, \mathtt{Sig}, \mathtt{Ver})$ for a digital signature.

In the above notation, KeyGen is a probabilistic algorithm, which takes a security parameter and which outputs a secret key for a symmetric-key cryptography, private/public key pair for an asymmetric-key cryptography, and a signing/validation key pair for a digital signature. Enc is a probabilistic algorithm that takes a plaintext and encryption key, i.e., a secret key for a symmetric-key cryptography and a public key for an asymmetric-key cryptography, and that outputs a ciphertext. Dec is a deterministic algorithm that takes a ciphertext and a decryption key, i.e., a secret key for a symmetric-key cryptography and a secret key for an asymmetric-key cryptography, and that outputs the resulting plaintext or the invalid ciphertext $\perp$. Sig is a probabilistic algorithm that takes a signing key and a document, and that outputs a signature. Ver is a deterministic algorithm that takes a verifying key, a document and a signature, and that outputs the validation result, i.e., valid or invalid.

3.2. Construction

The main idea for our construction is to use a one-time symmetric-key cryptography and a one-time signature. We adopt a hybrid construction with these techniques for read-write protection on a document. Let m be a document and we consider the following procedure:

$$c \leftarrow \mathtt{SE.Enc}_k(m) \text{ and then } \sigma \leftarrow \mathtt{DS.Sig}_s(c), \qquad (1)$$

where k is a secret key for SE and s is a signing key for DS. It is computationally hard to know any information of m from c unless the secret key k is given. In addition, we can generate a valid signature for any document if the signing key s is given, although it is infeasible to forge any signatures with respect to the signing key s without the knowledge of s. We use the former property to control read protection and the later one to control write protection.

Complete description of the proposed construction is shown in Figure 2. Note that we call this construction "Content Protection Scheme with Standard Cryptographic Techniques(CPS-SCT)". The above procedure is applied to each block m_i in a document M(see step 1 and 2 in the encryption algorithm). The secret key for encrypting m_i and signing key for generating σ_i is encrypted by encryption keys ek_j of an asymmetric-key cryptography PE, according to the access control list AC. Thanks to this procedure, if a user, who has a decryption key dk_j, has right to read an i-th block m_i, he/she can get the secret key k_i by the decryption result of e_j and he/she can read m_i(step 2 and 3-2 in the decryption algorithm). Also, if he/she has right to edit the i-th block, he/she can get the signing key s_i by decrypting e_j and can generate a valid signature for any messages. Therefore if he/she wants to edit the block m_i, he/she can regenerate a valid signature for a message m_i^+ and can edit the document M by replacing the original block m_i and σ_i with the edited ones m_i^+ and σ_i^+(see the update algorithm in Figure 2). Note that the step 3 in the encryption algorithm in Figure 2 is required for protecting the access control for the document M decided by a sender. The security of the proposed construction is discussed in the next subsection.

3.3. Security

We showed a concrete construction of a content protection scheme by combining a symmetric-key cryptography, an asymmetric-key cryptography and a digital signature. As we explained in the last subsection, this construction can control rights for both reading and writing a part of a document.

Formally, we have the following propositions for the confidentiality and integrity based on those definitions (see section 2.2). Note that the security notion of symmetric-key encryption scheme, asymmetric-key encryption scheme and digital signature schemes are referred in [2], [3] and [19].

Proposition 1: The proposed construction described in figure 2 has the property of the confidentiality, if the symmetric-key cryptography SE, the asymmetric-key cryptography PE and the signature scheme DS are respectively secure in the sense of IND-CPA, IND-CCA and EUF-CMA.

Proposition 2: The construction described in figure 2 has the property of the integrity if the proposed construction described in figure 2 has the property of the confidentiality, if the symmetric-key cryptography SE, the asymmetric-key cryptography PE and the signature scheme DS are respectively secure in the sense of IND-CPA, IND-CCA and EUF-CMA.

The sketch of the proofs for the above propositions are shown in Appendix.

4. Discussion

(1) Advantages

We can realize the proposed construction, CSP-SCT, by using well-known schemes, e.g., AES[15] with CBC, RSA-OAEP[17], ECDSA[14] and so on. This means that the proposed scheme can be easily implemented by a standard cryptographic library without any modification of cryptographic functions in the library. In addition, if users have key sets for an asymmetric-key cryptography PE and a digital signature scheme DS for some application, the users may reuse their key sets for an application of the proposed schemes.

(2) Improvement

The proposed construction, CSP-SCT, needs to generate the key pairs for each block of a document m_i and to resign when an i-th block is edited. This means that the proposed construction are somewhat inefficient from the viewpoint of computational cost. However, key pairs can be preliminarily

Key Generation:

Key Generation algorithm accepts a security parameter $\lambda = (\lambda_{\texttt{PE}}, \lambda_{\texttt{DS}}, \lambda_{\texttt{SE}})$ and generate key pairs (sk, vk) for a sender by $\texttt{DS.KeyGen}(\lambda_{\texttt{DS}})$ and a key pairs $(ek_1, dk_1), \ldots, (ek_u, dk_u)$ for recipients by $\texttt{PE.KeyGen}(\lambda_{\texttt{PE}})$.

Encryption:

Encryption algorithm accepts encryption keys $EK = (ek_1, \ldots, ek_u)$, sender's signing key sk, a plaintext $M = (m_1, \ldots, m_n)$ and an access control list $AC = (ac_1, \ldots, ac_u)$ $ac_i = (r_1^{(i)}, \ldots, r_n^{(i)}), r_j^{(i)} = \{\texttt{w}, \texttt{r}, \texttt{n}\}, (j = 1, \ldots, n)$. Note that, $\texttt{w}$, $\texttt{r}$ and $\texttt{n}$ respectively denote a right for a block, "can read and write", "can read but not write" and "can not read". Encryption algorithm processes the following procedure.

```
1. process the following for each $m_i$ $(i = 1, \ldots, n)$.
   1-1. generate a key $k_i$ by $\texttt{SE.KeyGen}(\lambda_{\texttt{SE}})$ and encrypt $m_i$ with $k_i$: $c_i \leftarrow \texttt{SE.Enc}_{k_i}(m_i)$.
   1-2. generate a key pair $(s_i, v_i)$ by $\texttt{DS.KeyGen}(\lambda_{\texttt{DS}})$ and generate a signature, $\sigma_i \leftarrow \texttt{DS.Sig}_{s_i}(i||c_i)$.
2. process the following for $ac_i = (r_1^{(i)}, \ldots, r_n)$ $(i = 1, \ldots, u)$.
   2-1. for $r_j^{(i)}$ $(j = 1, \ldots, n)$, if $r_j^{(i)} = \texttt{w}$ then $d_j^{(i)} \leftarrow k_i||s_i$, else if $r_j^{(i)} = \texttt{r}$ then $d_j^{(i)} \leftarrow k_i$, else if $r_j^{(i)} = \texttt{n}$
        then $d_j^{(i)} \leftarrow \texttt{NULL}$.
   2-2. encrypt $D_i = (d_1^{(i)}|| \ldots ||d_n^{(i)})$ with $ek_i$: $e_i \leftarrow \texttt{PE.Enc}_{ek_i}(D_i)$.
3. generate a signature $\sigma \leftarrow \texttt{DS.Sig}_{sk}(AC, e_1, \ldots, e_u, v_1, \ldots, v_n, \sigma_1, \ldots, \sigma_n)$
4. output the ciphertext $C = (AC, e_1, \ldots, e_u, v_1, \ldots, v_n, \sigma_1, \ldots, \sigma_n, c_1, \ldots, c_n, \sigma)$
```

Decryption:

Decryption algorithm accepts a decryption key dk_t, validation key vk and a ciphertext C. The decryption algorithm processes the following.

```
1. validate $(AC, e_1, \ldots, e_u, v_1, \ldots, v_n, \sigma_1, \ldots, \sigma_n, c_1, \ldots, c_n)$ by using a signature $\sigma$ and a validation key $vk$.
   If the validation is failed, then the decryption algorithm outputs $\perp$ and halts the process.
2. decrypt $e_t$ with $dk_t$.
3. refer to $ac_t = (r_1^{(t)}, \ldots, r_n^{(t)})$ and process the following for $c_i (i = 1, \ldots, n)$.
   3-1. validate $(i||c_i)$ by using the signature $\sigma_i$ and a validation key $v_i$. If the validation is failed, then the decryption
        algorithm outputs $\perp$ and halts the process.
   3-2. if $r_i^{(t)} = \texttt{w}$ or $\texttt{r}$, then decrypt $c_i$ with key $k_i$ given from step 2: $m_i^* \leftarrow \texttt{SE.Dec}_{k_i}(c_i)$. Otherwise, $m_i^* \leftarrow \texttt{NULL}$.
4. output $M^* \leftarrow (m_1^*, \ldots, m_n^*)$ as the decryption resulting.
```

Update:

Update algorithm accepts a decryption key dk_t, a ciphertext C and update information $U = (j, m_j^+)$ and processes the following.

```
1. validate $(AC, e_1, \ldots, e_u, v_1, \ldots, v_n, \sigma_1, \ldots, \sigma_n, c_1, \ldots, c_n)$ by using a signature $\sigma$ and a validation key $vk$.
   If the validation is failed, then the decryption algorithm outputs $\perp$ and halts the process.
2. validate $(i||c_i)$ $(i = 1, \ldots, n)$ by using the signature $\sigma_i$ and a validation key $v_i$. If the validation is failed, then
   the decryption algorithm outputs $\perp$ and halts the process.
3. decrypt $e_t$ with $dk_t$.
4. refer to $r_j^{(t)}$, and if $r_j^{(t)} \neq \texttt{w}$ then the update algorithm outputs $\perp$ and halts.
5. encrypt $m_j^+$ with a key $k_j$ given from step 3: $c_j^+ \leftarrow \texttt{SE.Enc}_{k_j}(m_j^+)$.
6. generate a signature $\sigma_j^+$ by using $s_j$ given from step 3: $\sigma_j^+ \leftarrow \texttt{DS.Sig}_{s_j}(j||c_j^+)$.
7. replace $c_j$ and $\sigma_j$ with $c_j^+$ and $\sigma_j^+$, respectively (the resulting ciphertext denotes $C^+$).
8. output $C^+$ as the update result.
```

Figure 2. Complete description of CPS-SCT(Content Protection Scheme with Standard Cryptographic Techniques)

generated before the document M is encrypted. In addition, if we adopt an on-line/off-line signature[8] for the proposed schemes, we can improve the computational cost for signing each block.

In addition, we may use an applied cryptographic technique instead of a standard cryptographic technique in the proposed construction. The aggregate signature proposed by D. Boneh, et. al.[6], is one of the examples. The aggregate signature scheme allows us to aggregate signatures, which are generated for distinct messages by several signers, as one signature. By using the aggregate signature scheme, we can reduce the ciphertext size of the proposed construction because we can aggregate signatures σ_i for blocks m_i as one signature Σ. More precisely, if we use the scheme proposed by D. Boneh et. al.[6], then the aggregate signature Σ is described as $\Sigma = \prod_{i=1,\ldots,n} \sigma_i$ (for the detail of the aggregate signature, see [6]). Note that if a recipient edits an i-th block c_i, he/she can know the signing key s_i for the block c_i and calculate the signature σ_i for the block c_i Therefore he/she can calculate the updated aggregate signature Σ^+ as $\Sigma^+ = \Sigma \cdot \sigma_i^+ / \sigma_i$, where σ_i^+ denotes the signature of the updated block c_i^+.

(3) Applications

Several applications are considered for the proposed schemes such as document creation by several users, outsourcing database and so on. We introduce a workflow system as an example application for the proposed schemes. Some workflow systems such as one-stop services are performed by several organizations. In these cases, it is hard to construct a trusted server for managing such a workflow system because each organization is independent on the others. In addition, an application form in the workflow system contains several documents that contains several kinds of personal information. Therefore, in the workflow system, reviewers check a necessary part of the application form and have to modify it for adding some comments or

Table 1. Experimental Results

(a) Encryption

Items		Time [ms]
For each block		477
	Enc. and Key Gen. by SE	0.3
	Key Generation for DS	464
	Signing by DS	12.7
For header		144
	Encryption by PE	108
	Signing by DS	35.6
Total		1,097

(b) Decryption

Items		Time [ms]
For header		17.1
	Verifying by DS	> 0.1
	Decryption by PE	17.1
For each block		> 0.1
	Verifying by DS	> 0.1
	Decryption by SE	> 0.1
Total		18.0

(c) Edit

Items		Time [ms]
For header		16.5
	Verifying by DS	16.5
	Decryption by PE	> 0.1
For each block		15.0
	Verifying by DS	> 0.1
	Re-encryption by SE	> 0.1
	Re-signing by DS	> 0.1
Total		31.5

Experimentation environments:

Computer:

PC: CPU: Intel®Pentium®4 2.7 GHz / Memory: 256MB RAM
OS: Microsoft®Windows®XP SP3
Language: Java(TM) Platform Standard Edition 6

Algorithms:

PE: Hybrid Encryption (RSA-KEM, AES with CBC mode and HMAC with SHA256)
DS: SHA256 with RSA
SE: AES with CBC mode
Key size: 1024-bit for RSA and 128-bit for AES

Target Document:

$M = (m_1, m_2)$, where m_1 and m_2 are randomly generated 1kB binary data.

$AC = (ac_1, ac_2), ac_1 = (\mathtt{w}, \mathtt{r}), ac_2 = (\mathtt{r}, \mathtt{w})^*$

* There are two recipients. One of the recipients can edit m_1 and can read m_2 but not edit. The other recipient can read m_1(but not edit) and can edit m_2.

We have 1000 experiments and show the average time of the experiments for each operation. The result of decryption and edit are the average time for the first user who can access the document M according to ac_1. The total time for the encryption includes the time for processing two blocks and two headers(for two recipients). Also, the total time for the decryption and the edit includes the time for processing two blocks and one header. Note that the re-encryption and re-signing for the edit were done for the first block by using randomly generated 1kB binary data.

correcting some items. By using the proposed schemes, a server that accepts the application form can control the read-write protection on the application form for each reviewer, even if the procedure of the workflow system is done by several independent organizations. In addition, the proposed schemes immunize fraudulence of an administrator because he/she can not read or edit an item in a document in a workflow system if he/she is not given rights to read or edit it by the sender.

5. Implementation

We have an experiment to investigate the performance of each function in the proposed scheme, CPS-SCT. For this experiment, we implement the proposed construction with standard cryptographic schemes: AES[15] with CBC mode[16] for SE, RSA signature with PKCS #1.5[17] for DS and a hybrid encryption[10] for PE. More precisely, we use RSA with SHA256[13] for RSA signature and RSA-KEM, AES with CBC mode and HMAC with SHA256 for DEM in the hybrid encryption. Note that 1024-bit keys and 128-bit keys are used for RSA and AES, respectively.

The experimental result and environments are shown in Table 1. The experimental result shows the processing time of the encryption is over 1 second. However, as we pointed out in the last section, the key generation for DS, which is the major part in the encryption, can be preliminary processed and therefore the processing time of the encryption is reduced substantially. If we perform the key generation before the encryption, the total time of the encryption takes about 170ms. In addition, if we adopt an elliptic curve based signature scheme, e.g., ECDSA[14], for the signature scheme DS, the time consumption for the key generation is reduced compared to RSA based signature scheme, e.g., RSA with SHA256[17].

Note that we consider no file format for our experiment, i.e., the experimental result does not include a performance of a parser. For applying our implementation to a specific file format, e.g., XML, PDF, and so on, we need to use a parser for the file format. The performance of such a parser is out of our experimental result.

6. Conclusion

In this paper, we proposed cryptographic schemes that achieve read-write protection in a document, which we call content protection schemes(CPSs). Since the CPSs enable us to control read-write protection only by a decryption key, the schemes can promote proper use of digital documents even if we could not manage them on a server.

The concrete construction for CPSs, which we proposed in section 3, consists of only standard cryptographic techniques: a symmetric-key cryptography, an asymmetric cryptography and a digital signature. This makes us easy to implement the proposed construction and users does not need to prepare special key sets for an application of CPSs. We showed the experimental results of the proposed construction in section 4.

In addition, the proposed construction might be improved by using an applied cryptographic scheme, e.g., ID-based encryption schemes[5], Attribute-Based encryption schemes[4] and so on, so that the improved construction inherits the property of the applied cryptographic schemes. To achieve the improvement and more efficient construction is one of our future works.

References

[1] G. Ateniese, D. H. Chou, B. Medeiros and G. Tsudik, “Sanitizable Signatures”, Computer Security – ESORICS 2005, 10th European Symposium on Research in Computer Security, Lecture Notes in Computer Science, Vol. 3679(LNCS 3679), pp. 159-177, 2005.

[2] M. Bellare, A. Desai, E. Jokipii and P. Rogaway, “A Concrete Security Treatment of Symmetric Encryption”, Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, pp. 394-405, 1997.

[3] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway, “Relations Among Notions of Security for Public-Key Encryption Schemes”, Advances in Cryptology – CRYPTO’98, Lecture Note in Computer Science Vol. 1462 (LNCS 1462), pp.26-46, 1998.

[4] J. Bethencourt, A. Sahai, B. Waters, “Ciphertext-Policy Attribute-Based Encryption”, Proceedings of the 2007 IEEE Symposium on Security and Privacy, pp.321-334, 2007.

[5] D. Boneh, M. K. Franklin, “Identity-Based Encryption from the Weil Pairing”, Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology(CRYPTO 2001), Lecture Notes In Computer Science, Vol. 2139, pp. 213 - 229, 2001.

[6] D. Boneh, C. Gentry, H. Shacham and B. Lynn, “Aggregate and verifiably encrypted signatures from bilinear map”, Proceedings of Advances is Cryptology – Eurocrypt’03, Lecture Note on Computer Science, vol. 2656(LNCS 2656), pp. 416-432, 2003.

[7] S. Canard, F. Laguillanumie and M. Milbau, “Trapdoor Sanitizable Siganture and Their Application to Content Protection”, The Proceedings of the 6th International Conference on Applied Cryptography and Network Security (ACNS 2008), Lecture Note on Computer Science, Vol.037 (LNCS 5037), pp.258-276, Springer, 2008.

[8] S. Even, O. Goldreich and S. Micali, “On-Line/Off-Line Digital Signatures”, Journal of Cryptology, pp. 35-67, 1996.

[9] Y. Hatano, K. Miyazaki and T. Kaneko, “A Study on Extended Multi-Recipient Encryption : Security Notion and Constructions”, IEICE Technical Report, Vol.107, No.209((ISEC2007-88), pp. 107-114, 2007. (In Japanese)

[10] International Organization for Standardization(ISO), “ISO/IEC 18033-1:2005 Information technology – Security techniques – Encryption algorithms – Part 1: General”, 2005.

[11] K. Miyazaki, S. Susaki, M. Iwamura, T. Matsumoto, R. Sasaki and H. Yoshiura, “Digital Document Sanitizing Problem”, Technical report of IEICE, Vol.103, No.195(ISEC2003-20), pp. 61-67, 2003. (in Japanese)

[12] R. Johnson, D. Molnar, D. Song and D. Wagner, “Homomorphic Signature Schemes”, Topics in Cryptology - CT-RSA 2002, Lecture Notes in Computer Science, Vol. 2271(LNCS 2271), pp.204-245, 2002.

[13] Federal Information National Institute of Standards and Technology (NIST), “Specification for the Secure Hash Standard”, Federal Information Processing Standards Publication 180-3(FIPS 180-3), Octorber, 2008.

[14] Federal Information National Institute of Standards and Technology (NIST), “Draft Digital Signature Standard (DSS)”, Draft Federal Information Processing Standards 186-3 (DRAFT FIPS 186-3), November, 2008.

[15] Federal Information National Institute of Standards and Technology (NIST), “Specification for the Advanced Encryption Standard (AES)”, Federal Information Processing Standards Publication 197(FIPS-197), November 26, 2001.

[16] Federal Information National Institute of Standards and Technology (NIST), “Recommendation for Block Cipher Modes of Operation: Methods and Techniques”, NIST Special Publication 800-38A, November, 2001.

[17] RSA Laboratories, “PKCS #1: RSA Cryptography Standard”, Public-Key Cryptography Standards (PKCS), 2002.

[18] A. Saito, Y. Yamada and K. Iwamura, “The Proposal of E-signature System using Sanitizable Signature for Editional Contents”, IPSJ SIG Technical Report, Vol. 2007, No. 126(2007-CSEC-039), pp. 49-54, 2007. (in Japanese)

[19] R. Steinfeld, L. Bull and Y. Zheng, “Content Extraction Signatures”, Information Security and Cryptology–ICISC 2001: 4th International Conference, Seoul, Korea, December 6-7, 2001 : Proceedings (Lecture Notes in Computer Science, Vol. 2288(LNCS 2288), pp.285-304, 2001.

[20] M. A. Rahaman, Y. Roudier and A. Schaad, “Distributed Access Control For XML Document Centric Collaborations”, The proceedings of the 12th International IEEE Enterprise Distributed Object Computing Conference (EDOC 2008), pp.267-pp.276, 2008.

[21] T. H. Yuen, W. Susilo, J. K. Liu and Y. Mu, “Sanitizable Signatures Revisited”, 7th International Conference of Cryptology and Network Security(CANS 2008), Lecture Notes in Computer Science, Vol. 5339(LNCS 5339), pp. 80-97, 2008.

Appendix

1. Security proof

In this section, we give the sketch of the formal security proofs for the confidentiality and the integrity of the proposed construction described in figure 2. The security notion of symmetric-key encryption scheme, asymmetric-key encryption scheme and digital signature schemes are referred in [2], [3] and [19].

1.1. Sketch of proof for proposition 1.

Let $\mathcal{A}$ and $\mathbf{Adv}(\mathcal{A})$ be an adversary for CPSs in the sense of the confidentiality and the advantage of the adversary $\mathcal{A}$. First we modify the original game in the definition of the confidentiality, which we call Game 0 so that all queries for decryption oracle are rejected, and we call this modified game Game 1. For this modification, we can easily construct an adversary that break the integrity of the proposed construction. Let $\mathcal{A}^{\mathtt{INT}}_{\mathtt{CPS}}$ and $\mathbf{Adv}(\mathcal{A}^{\mathtt{INT}}_{\mathtt{CPS}})$ be an adversary that break the proposed construction in the sense of the integrity and the advantage of the adversary $\mathtt{A}^{\mathtt{INT}}_{\mathtt{CPS}}$. We have

$$|\Pr[Succ(\text{Game } 0)] - \Pr[Succ(\text{Game } 1)]| \leq \mathbf{Adv}(\mathcal{A}^{\mathtt{INT}}_{\mathtt{CPS}}), \quad (2)$$

where $Succ(\text{Game } 0)$ and $Succ(\text{Game } 1)$ denote the events that an adversary $\mathcal{A}$ succeeds Game 0 and Game 1, respectively.

Next we further modify the game 1 so that e_i ($i = 1, \ldots, w$) takes a random value e_i^*, and we call this modified game as game 2. For this modification, we can construct an adversary $\mathcal{A}_{\mathtt{PE}}$ that breaks an asymmetric-key encryption scheme in the sense of IND-CCA2. Let $\mathbf{Adv}(\mathcal{A}_{\mathtt{PE}})$ be the advantage of the adversary $\mathcal{A}_{\mathtt{PE}}$. Then we have

$$|\Pr[Succ(\text{Game 1})] - \Pr[Succ(\text{Game 2})]| \leq w \cdot \mathbf{Adv}(\mathcal{A}_{\mathtt{PE}}) \qquad (3)$$

where $Succ(\text{Game 2})$ denotes the event that the adversary $\mathcal{A}$ succeeds Game 2.

Finally, for the game 2, we can construct an adversary $\mathcal{A}_{\mathtt{SE}}$ that breaks a symmetric-key encryption scheme in the sense of IND-CPA. Let $\mathbf{Adv}(\mathcal{A}_{\mathtt{SE}})$ be the advantage of the adversary $\mathcal{A}_{\mathtt{SE}}$. We have

$$\left|\Pr[Succ(\text{Game 2})] - \frac{1}{2}\right| \leq \mathbf{Adv}(\mathcal{A}_{\mathtt{SE}}). \qquad (4)$$

From the inequalities (2), (3) and (4), we have

$$\mathbf{Adv}(\mathcal{A}) \leq \mathbf{Adv}(\mathcal{A}^{\mathtt{INT}}_{\mathtt{CPS}}) + w \cdot \mathbf{Adv}(\mathcal{A}_{\mathtt{PE}}) + \mathbf{Adv}(\mathcal{A}_{\mathtt{SE}}).$$

The advantages of $\mathbf{Adv}(\mathcal{A}_{\mathtt{PE}})$, $\mathbf{Adv}(\mathcal{A}_{\mathtt{SE}})$ and $\mathbf{Adv}(\mathcal{A}^{\mathtt{INT}}_{\mathtt{CPS}})$ are negligible, if PE, SE and CPS are an IND-CCA2 secure asymmetric encryption, an IND-CPA secure symmetric encryption and has the integrity property. Therefore, $\mathbf{Adv}(\mathcal{A})$ is negligible and we can say that the proposed construction has the property of the confidentiality. Note that $\mathbf{Adv}(\mathcal{A}^{\mathtt{INT}}_{\mathtt{CPS}})$ is discussed in the proposition 2. □

1.2. Sketch of proof of proposition 2.

A ciphertext of the proposed construction consists of two part besides a signature σ. One is $\phi = (AC, e_1, \ldots, e_u, v_1, \ldots, v_n)$ which are signed by a signing key sk of a sender and the other is $\mu = (c_1, \ldots, c_n, \sigma_i, \ldots, \sigma_n)$, where each c_i is signed by randomly generated signing keys s_i and σ_i is a signatures of c_i: $C = (\phi, \mu, \sigma)$.

Let $\mathcal{A}$ and $\mathbf{Adv}(\mathcal{A})$ be an adversary for CPSs in the sense of the integrity and the advantage of the adversary $\mathcal{A}$. First we modify the original game in the definition of the integrity, which we call Game 0, so that e_i ($i = 1, \ldots, u$) takes a random value e_i^*, and we call this modified game Game 1. For this modification, we can construct an adversary $\mathcal{A}_{\mathtt{PE}}$ for the asymmetric-key encryption scheme in the sense of IND-CCA by using an adversary $\mathcal{A}$. Then we have

$$|\Pr[Succ(\text{Game 0}) - \Pr[Succ(\text{Game 1})]| \leq w \cdot \mathbf{Adv}(\mathcal{A}_{\mathtt{PE}}). \qquad (5)$$

Note that $Succ(\text{Game 0})$ and $Succ(\text{Game 1})$ denote the event that the adversary $\mathcal{A}$ succeeds Game 0 and Game 1, respectively, and $\mathbf{Adv}(\mathcal{A}_{\mathtt{PE}})$ denotes an advantage of the adversary $\mathcal{A}_{\mathtt{PE}}$.

In the modified game Game 1, we show the following inequality.

$$\Pr[Succ(\text{Game 1})] \leq (n+1) \cdot \mathbf{Adv}(\mathcal{A}_{\mathtt{DS}}])$$

In order to show this, we construct an adversary $\mathcal{A}$ for the propose construction succeeds to the modified game Game 1, then an attacker $\mathcal{A}_{\mathtt{DS}}$ for a standard digital signature scheme for at least one of these two parts, ϕ and μ, exists with non-negligible probability.

For Game 1, exploiting ϕ in ciphertexts C, which is signed by a signing key sk, we can construct $\mathcal{A}_{\mathtt{DS}}$ by using the adversary $\mathcal{A}$, and we have

$$\begin{aligned}\Pr[Succ(\mathcal{A}_{\mathtt{DS}})] &\geq \Pr[Succ(\text{Game 1}) \wedge \mathtt{NewPhi}] \\ &= \Pr[Succ(\text{Game 1})] \\ &\quad - \Pr[Succ(\text{Game 1}) \wedge \neg\mathtt{NewPhi}], \qquad (6)\end{aligned}$$

where $Succ(\mathcal{A}_{\mathtt{DS}})$ are events that the adversary $\mathcal{A}_{\mathtt{DS}}$ succeeds to produce a valid signature. In addition, NewPhi denotes an event that ϕ has not been queried to the signing oracle for the digital signature scheme.

Since a ciphertext of the proposed construction is $C = (\phi, \mu, \sigma)$, $Succ(\text{Game 1}) \wedge \neg\mathtt{NewPhi}$ means that $\phi = \phi^{\#}$ and $\mu \neq \mu^{\#}$, where $\phi^{\#}$ and $\mu^{\#}$ are the ones in a query ciphertext $C^{\#}$.

For $Succ(\text{Game 1}) \wedge \neg\mathtt{NewPhi}$, we construct another adversary $\mathcal{B}_{\mathtt{DS}}$ for a standard digital signature scheme by using the adversary $\mathcal{A}$ and then we have

$$\Pr[Succ(\mathcal{B}_{\mathtt{DS}})] \geq \left(\frac{1}{n}\right) \cdot \Pr[Succ(\text{Game 1}) \wedge \neg\mathtt{NewPhi}] \qquad (7)$$

From the inequalities (6) and (7), we have

$$\Pr[Succ(\text{Game 1})] \leq \Pr[\mathcal{A}_{\mathtt{DS}}] + n \cdot \Pr[\mathcal{B}_{\mathtt{DS}}]$$

From the above, since the upper bound of $\Pr[\mathcal{B}_{\mathtt{DS}}]$ must be the same as the one of $\Pr[\mathcal{A}_{\mathtt{DS}}]$, we have

$$\Pr[Succ(\text{Game 1})] \leq (n+1) \cdot \mathbf{Adv}(\mathcal{A}_{\mathtt{DS}}). \qquad (8)$$

From the inequalities (5) and (8), we have

$$\mathbf{Adv}(\mathcal{A}) \leq (n+1) \cdot \mathbf{Adv}(\mathcal{A}_{\mathtt{DS}}) + w \cdot \mathbf{Adv}\left(\mathcal{A}_{\mathtt{PE}}\right).$$

This shows that if an adversary $\mathcal{A}$ breaks the property of the integrity of the proposed construction then an adversary $\mathcal{A}_{\mathtt{DS}}$ with non-negligible advantage or $\mathcal{A}_{\mathtt{PE}}$ with non-negligible advantage exists. □

Intel and Pentium are registered trademarks of Intel Corporation in the U.S. and other countries. Microsoft and Windows are registered trademarks of Intel Corporation in the U.S. and other countries. Java is a trademark of Sun Microsystems, Inc. in the United States and other countries.