Cryptographic Approch for Workflow Systems

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

[1] Systems Development Laboratory, Hitachi, Ltd., 292, Yoshida-cho, Totsuka-ku, Yokohama-shi, Kanagawa-ken, 244-0817,
[2] Tokyo University of Science, 2641, Yamazaki, Noda-shi, Chiba-ken, 278-8510,
e-mail: {yasuo.hatano.bn, kunihiko.miyazaki.zt}@hitachi.com, kaneko@ee.noda.tus.ac.jp

Abstract This paper proposes encryption schemes to enforce the order of the procedure in a workflow system. In workflow systems, it is important to perform a procedure according to an order that is given by some regulation. In addition, it is desired that each reviewer checks a necessary part of a document to admit the application because the document sometime contains privacy information, e.g., name, birthday, income and so on. There is a procedure in a workflow system which it is sufficient to pass if one of two reviewers admits the document. More generally, there is a procedure in a workflow system that can be passed if k of n reviewers admit the document, which we call a "threshold procedure". By applying a cryptographic technique, e.g., a multiple encryption and secret sharing, this paper gives a method to realize various procedures in workflow systems, i.e., controlling the order of reviewers, disclosing a part of document selectively, and a threshold procedure. Conventional workflow systems control their procedure by a server, which we consider a trusted one. This implies that an administrator might tamper a procedure. The proposed schemes help us to enforce a procedure even if he/she is not trustworthy.

Please use the following format when citing this chapter:

Hatano, Y., Miyazaki, K., Kaneko, T., 2008 in IFIP International Federation for Information Processing, Volume 286; Towards Sustainable Society on Ubiquitous Networks, eds. Oya, M., Uda, R., Yasunobu, C., (Boston: Springer), pp. 75–86.

1 Introduction

1.1 Background

Workflow systems provide an automated procedure for a business process and they are used to computerize and to automate various procedures in a company. Recent accounting scandals in various countries have resulted in the passage of several laws establishing or enhancing standards for corporate financial reporting and record keeping. Typical examples are the Sarbanes-Oxley (SOX) Act in the United State[1] and the Japanese version of that act, the Financial Commodities Exchange Act[2]. They require companies to establish internal control and workflow systems are one of the important components to establish effective internal controls, because workflow systems give us a proper method to perform a procedure of a business process. In a procedure in a business process, various reviewers need to check a submitted document from an employee (applicant) and it is important to control the reviewing order which is defined by the business process.

To control the reviewing order is required not only for a business process in a company but also for a procedure performed through several corporations. In a one-stop service of public institutions, a procedure is sometime performed through several offices. For instance, if we apply a procedure to buy a car in Japan by a one-stop service, the procedure needs to access systems of several offices because, in the physical procedure, we first have to submit a certificate of a parking area to a police office and then submit an application to issue a car number to Distinct Transport Bureau. Other examples of procedures performed through several corporations are the one to issue a credit card and to claim a payment of insurance. To issue a credit card, the credit card company needs to check an applicant by credit facilities and then issues a credit card after checking the application in the card company.

Workflow systems are usually constructed with a server, which we consider a trusted one, and the server manages reviewers, the document and the status of an application in a procedure and controls the order of the procedure. In this paper, we show a cryptographic technique to support a workflow system. We first propose a basic construction to enforce the order of a procedure by adopting a multiple encryption and then we propose enhanced constructions for threshold procedures and for partial disclosure of a document. Note that we call procedures that can be passed if k of m reviewers admit the application "threshold procedures".

To realize a threshold procedure has several advantages for workflow systems. One advantage is that, by combining basic constructions and threshold procedures, we can construct any kinds of procedures for workflow systems even if the order of a procedure is complicated. Another advantage is that a threshold procedure allows us to make a pass for a substitute reviewer. Even if a regular reviewer is absent, e.g., a manager travels on business, a substitute reviewer, e.g., an assistant manager, can admit a document by a threshold procedure.

Moreover, the partial disclosure of a document in a procedure is important because some documents in workflow systems contain sensitive information, e.g.,

name and birthday, and it is desired that such information is disclosed to only proper reviewers. For instance, a credit card application contains sensitive information like a name, birthday, annual income, bank account number, PIN number and so on. As we mentioned in the above, in a procedure to issue a credit card, the card company needs to check an application by credit facilities. The card company, however, should not submit the whole information for credit facilities. In this case, the card company should hide the PIN number for credit facilities and should hide the income and PIN number for a bank. In this paper, we give a method to realize threshold procedures by using a secret sharing[15] and also show partial disclosure of a document in a procedure by adopting hybrid encryption schemes[5, 8].

As we mentioned in the above, conventional workflow systems use a server, which we consider a trusted one, and the server controls the order of a procedure. Such systems, however, imply that an administrator of a server might tamper a procedure and associated log entries that are important evidence for a procedure. Hence if the administrator is not trustworthy, nobody knows whether the procedure have been done properly or not. This means that internal control does not achieve if we do not trust an administrator of a workflow system. In addition, if a workflow system is collaborated with several corporations, e.g., one-stop services and credit card applications, it is hard to manage a whole procedure because there are several administrators. Using the proposed method, we can construct a workflow system that strongly protects the order of a procedure by a cryptographic technique. This enables us to enforce a procedure even if an administrator is not trustworthy. In addition, the proposed schemes facilitate constructing a procedure collaborated with several corporations because the order of the procedure can be decided only by an applicant or a reception.

1.2 Related Work

As we mentioned in the above, we adopt a multiple encryption to control the reviewing order of a procedure. A multiple encryption is used for improving security[16]. Encrypting a plaintext by several encryption algorithms, information of the plaintext will not be leaked even if one of the encryption algorithms is broken.

Another application of multiple encryptions is “onion routing”[13], which is a technique to hide receiver’s information in a network communication. When Alice sends a message to Bob by using onion routing, she first chooses several routers and encrypts traffic information to Bob such that the received router can know only information about the next router. Since each router can know only the information about the next router, Alice can hide the information about the receiver Bob that she wants to send the message to.

Proxy re-encryption schemes[9] are methods that allow proxies to transform a ciphertext which has been encrypted for one party, so that it may be decrypted by another. Dodis proposed two framework of proxy re-encryption schemes, called “unidirectional proxy re-encryption” and “bidirectional proxy re-encryption”, and

proposed using a multiple encryption as a simple construction for a unidirectional proxy encryption[9].

The goals of related works introduced in the above are not the control of the reviewing order. To construct a workflow system, we propose encryption schemes that enable us to control an order of reviewers and show enhanced schemes to perform a threshold procedure and to disclose a necessary part of a document in a procedure. Using the proposed methods, we can construct various procedures with the order control by a cryptographic protection.

2 Preliminary

In this paper, we use the following notation.

- u_i : i-th reviewer.
- M_i : Plaintext browsed by i-th reviewer.
- C_i : Ciphertext received by i-th reviewer.
- n : Number of reviewer.
- *hash* : Cryptographic hash function.
- *Concat* : Concatenation function.
- *Parse* : Parser.
- `KeyGen` : Key generation algorithm.
- `Enc` : Encryption algorithm.
- `Dec` : Decryption algorithm.
- $\mathtt{SE} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$: Symmetric key cryptosystem.
- $\mathtt{PE} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$: Asymmetric key cryptosystem.

Note that, `KeyGen` is a probabilistic algorithm, which takes a security parameter and which outputs a secret key for symmetric key cryptosystems or private/public key pair for asymmetric key cryptosystems. `Enc` is a probabilistic algorithm that takes a plaintext and encryption key, i.e., a secret key for symmetric key cryptosystems and a public key for asymmetric key cryptosystems, and that outputs a ciphertext. `Dec` is a deterministic algorithm that takes a ciphertext and a decryption key, i.e., a secret key for symmetric key cryptosystems and a private key for asymmetric key cryptosystems, and that outputs the resulting plaintext or the invalid ciphertext $\perp$.

In the following, we use the notation `ES.Alg` to denote the algorithm `Alg` of `ES`. For instance, `SE.KeyGen` denotes the key generation algorithm `KeyGen` of a symmetric cryptosystem `SE`.

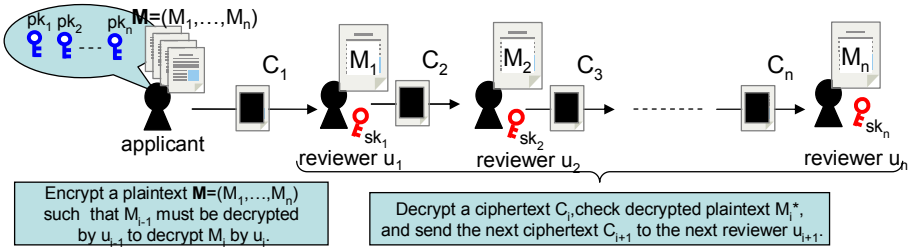

Fig. 1 Cryptographic Approach for Workflow System

3 Cryptographic Approach for Workflow Systems

3.1 Encryption Schemes with Reviewing Order Control

In this paper, we propose encryption schemes that enable us to enforce a reviewing order (see Fig. 1). In the proposed schemes, a sender(applicant) designates several reviewers and encrypts a plaintext $\mathbf{M} = (M_1, \ldots, M_n)$ by public keys of reviewers such that the reviewer $u_i (i = 2, \ldots, n)$ can only read the plaintext M_i after u_{i-1} read the plaintext M_{i-1}.

A reviewer u_i receives her/his ciphertext C_i and decrypts it with her/his private key sk_i. The decryption algorithm outputs the ciphertext C_{i+1} for the next reviewer u_{i+1} besides the decrypted plaintext M_i^* and the reviewer u_i reads M_i^* and sends C_{i+1} to the next reviewer u_{i+1}. Note that we call the proposed schemes as "Encryption Schemes with Reviewing Order Control (ESROC)" and that denote this as $\mathtt{ESROC} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$. We describe the algorithm of ESROC in the following.

Key Generation Algorithm `KeyGen`**:**
Probabilistic algorithm that takes a security parameter and that outputs a set of public/private key pairs (PK, SK). Note that $PK = (pk_1, \ldots, pk_n)$ and $SK = (sk_1, \ldots, sk_n)$ respectively denote a set of public keys and private keys and (pk_i, sk_i) denotes a public/private key pair for reviewer u_i.

Encryption Algorithm `Enc`**:**
Probabilistic algorithm which accepts a plaintext $\mathbf{M} = (M_1, \ldots, M_n)$ and a set of encryption keys PK and which outputs a ciphertext C_1 for the first reviewer u_1.

Decryption Algorithm `Dec`**:**
Deterministic algorithm which accepts a ciphertext C_i for the i-th reviewer u_i and the reviewer's private key sk_i and which outputs the decrypted plaintext M_i^* and the ciphertext C_{i+1} for the next reviewer u_{i+1}. The decryption algorithm outputs invalid ciphertext $\perp$ if the decryption process fails.

The difference from conventional encryption schemes and the proposed schemes is in the decryption algorithm `Dec`. The decryption algorithm of conventional en-

cryption schemes usually output only the resulting plaintext, but the decryption algorithm in the proposed encryption schemes outputs the ciphertext for the next reviewer besides the resulting plaintext.

3.2 Security Requirements

The proposed schemes are required that a ciphertext is decrypted according to the order that is defined by applicant who encrypts a plaintext. Moreover, the decrypted plaintext M_i^* can be read by only the reviewer u_i. Therefore the proposed schemes must have the following properties.

- Order Robustness
  It is infeasible to cheat the reviewing order for any probabilistic polynomial time (PPT) adversaries. More precisely, it is infeasible for any PPT adversaries to compute a ciphertext $C_i'(\neq C_i)$ which generates C_{i+1} by $\mathtt{ESROC.Dec}_{sk_i}(C_i)$.
- Confidentiality
  It is infeasible for any PPT adversaries to know any information about the plaintext M_i. More precisely, for a plaintext pair $\mathbf{M}^{(0)}$ and $\mathbf{M}^{(1)}$, where $\mathbf{M}^{(b)} = (M_1, \ldots, M_i^{(b)}, \ldots, M_n)$ $(b = \{0,1\})$, it is infeasible for any PPT adversaries to decide which plaintext, $\mathbf{M}^{(0)}$ or $\mathbf{M}^{(1)}$, is encrypted.

The notion of confidentiality is equivalent to indistinguishability (IND) notion for public key encryption schemes[6]. In addition, the notion of order robustness contains the notion of plaintext awareness (PA), which is another security notion for public key encryption schemes[6]. Therefore, from the result in [6], a secure encryption as the proposed schemes implies a secure public key encryption in the sense of IND-CCA2 (INDistinguishability against adaptive chosen ciphertext attacks). It should be noted that, for a secure encryption as the proposed schemes, it is not enough to be just an IND-CCA2 secure encryption because an IND-CCA2 secure public key encryption is not always a PA secure public key encryption although an public key encryption which is PA and IND-CPA secure is an IND-CCA2 secure public key encryption (see [6]).

4 Basis Construction

4.1 Construction

In this section, we propose a basic construction for ESROC. The proposed method in this section, we adopt the conventional asymmetric key cryptosystem and construct an ESROC by using a multiple encryption.

The encryption algorithm of the basic construction encrypts the i-th plaintext M_i with a ciphertext C_{i+1}, which is a ciphertext for the next reviewer u_{i+1}, and outputs a ciphertext C_{i-1} for the previous reviewer u_{i-1}. The decryption algorithm decrypts a ciphertext C_i by the private key sk_i of the i-th reviewer. The output of the decryption algorithm consists of two components: one is the decrypted plaintext M_i^* and the other is the ciphertext C_{i+1} for the next reviewer u_{i+1}. We show this construction in Fig. 2[1].

As shown in Fig. 2, in the basic construction, the number of reviewers increases size of a target ciphertext. Therefore, in practice, hybrid encryption schemes, e.g., [5, 8], is suitable to construct a basic construction.

$\mathtt{ESROC.KeyGen}(\lambda, n)$
for $i = 1$ to n do
$(pk_i, sk_i) \leftarrow \mathtt{PE.KeyGen}(\lambda)$;
end for;
output (PK, SK);

$\mathtt{ESROC.Enc}(\mathbf{M}; pk_1, \ldots, pk_n)$
$C_{n+1} \xleftarrow{R} \{0,1\}^L$;
for $i = n$ to 1 do
$C_i \leftarrow \mathtt{PE.Enc}(C_{i+1}||M_i; pk_i)$;
end for;
output C_1;

$\mathtt{ESROC.Dec}(C_i; sk_i)$;
$C \leftarrow \mathtt{PE.Dec}(C_i; sk_i)$;
$(C_{i+1}||M_i^*) \leftarrow Parse(C)$
output (C_{i+1}, M_i^*);

Fig. 2 Basic Construction

4.2 Security

A ciphertext C_i is generated from a plaintext M_i and a ciphertext C_{i+1} by the encryption algorithm of an asymmetric key cryptosystem PE.Enc. If the asymmetric key cryptosystem PE is secure enough, i.e., it is PA and IND-CPA secure(see [6]), we can say that the basic construction described in this section has the property of order robustness and confidentiality. Note that, even if an adversary know some private keys except the one for a target block i, the basic construction is secure.

[1] More generally, each reviewer can use different asymmetric key cryptosystems in a basic construction. Note that if the security of an asymmetric key cryptosystem depends on the message length, the security parameter must be carefully chosen.

5 Enhanced Constructions

5.1 Construction for Threshold Procedure

5.1.1 Improvement of the basic construction

There are some procedures in a workflow system such that several reviewers check a document and the order of them is not defined. For instance, there are two reviewers, $u^{(1)}$ and $u^{(2)}$ in a procedure and both reviewer have to admit the document but $u^{(1)}$ may admit before $u^{(2)}$ and $u^{(2)}$ may admit before $u^{(1)}$. In addition, there is a case where it is sufficient to pass a procedure if one of two reviewers admits the document. More generally, those procedures is described as the one that is passed if k of m reviewers admit the document and we call such procedures as (m,k) threshold procedures. By using threshold procedures, we can construct various procedures even if a procedure is complicated. In this subsection, we enhance the basic construction for (m,k) threshold procedures.

To improve the basic construction for a threshold procedure, we apply the Shamir's secret sharing[15] to the basic construction. We show the proposed construction for an (m,k) threshold procedure in Fig. 4. Note that, in Fig. 4, i-th reviewers in the (m,k) threshold procedure denotes $u_i^{(1)},\ldots,u_i^{(m)}$ and the private and pubic key of a reviewer $u_i^{(j)}(j=1,\ldots,m)$ denote $sk_i^{(j)}$ and $pk_i^{(j)}$, respectively. The set of reviewers, whose ciphertexts $C_i^{(x)}(x=1,\ldots,m)$ are received by reviewer u_{i+1}, denotes $\mathscr{U}_i \subseteq \{u_i^{(1)},\ldots,u_i^{(m)}\}$, where the number of elements in $\mathscr{U}_i$ is greater than or equal to k. Although we adopt a secret sharing for a threshold procedure, we may adopt all-or-nothing transform (AONT)[14] instead of secret sharing if a procedure requires that all of m reviewers admit the document.

5.1.2 Security

As we showed in Fig. 4, we divide an $(i+1)$-th ciphertext $C_{(i+1)}$ into m ciphertexts $C_{i+1}^{(1)},\ldots,C_{i+1}^{(m)}$ by using a secret sharing. Because of the property of secret sharing schemes, the $(i+1)$-th reviewer can recover the ciphertext $C_{(i+1)}$ if and only if he/she receives at least k ciphertexts in m ciphertexts, $C_{i+1}^{(1)},\ldots,C_{i+1}^{(m)}$. Since a ciphertext $C_{i+1}^{(j)}$ is given by the reviewer $u_{i+1}^{(j)}$, we can realize an (m,k) threshold procedure.

As we showed in Fig. 4, all of ciphertexts $C_i^{(j)}$, $C_{i+1}^{(j)}$ and C_{i+2} around a threshold procedure are obtained by decrypting the previous ciphertext $C_{i-1}^{(j)}$, $C_i^{(j)}$ and C_{i+1}, respectively. Therefore, from the same reason of the basic construction, if the asymmetric key cryptosystem PE is secure enough, the proposed construction in Fig. 4 is also secure.

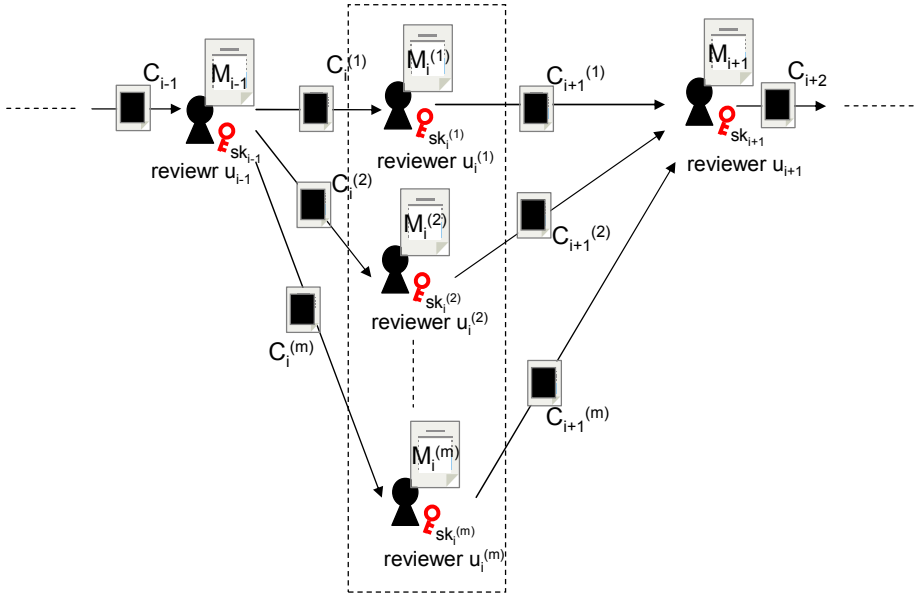

Fig. 3 Threshold Procedure

[Modification of Encryption Algorithm]

(1) $j = i+1$

$a_0 \leftarrow C_j$;

for $l = 1$ to k do $a_l \overset{R}{\leftarrow} \{0,1\}^*$; end for;

for $x = 1$ to m do

$C_j^{(x)} \leftarrow f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_k x^k$;

$C_{j-1}^{(x)} \leftarrow \mathtt{PE.Enc}(C_j^{(x)} || M_{j-1}^{(x)}, pk_j^{(x)})$;

end for;

(2) $j = i$

$C_j \leftarrow (C_{j-1}^{(1)} || C_{j-1}^{(2)} || \ldots || C_{j-1}^{(m)})$;

$C_{j-1} \leftarrow \mathtt{PE.Enc}(C_j || M_{j-1}, pk_{j-1})$;

[Modification of Decryption Algorithm]

(1) $j = i-1$

$C \leftarrow \mathtt{PE.Dec}(C_j, sk_j)$;

$(C_{j+1}^{(1)} || C_{j+1}^{(2)} || \ldots || C_{j+1}^{(m)} || M_j) \leftarrow Parse(C)$

output $C_{j+1}^{(1)}, C_{j+1}^{(2)}, \ldots, C_{j+1}^{(m)}, M_j$;

(2) $j = i$

$C \leftarrow \mathtt{PE.Dec}(C_j^{(x)}, sk_j)$;

$(C_{j+1}^{(x)} || M_j^{(x)}) \leftarrow Parse(C)$;

output $C_{j+1}^{(x)} || M_j^{(x)}$;

(3) $j = i+1$

$$C_j \leftarrow \sum_{x \in \mathscr{U}_j} \left(C_j^{(x)} \prod_{z \in \mathscr{U}_j, z \neq x} \frac{-z}{x-z} \right);$$

$C \leftarrow \mathscr{D}(C_j, sk_j)$;

$(C_{j+1} || M_j) \leftarrow Parse(C)$;

output $C_{j+1} || M_j$;

Fig. 4 Modification for Threshold Procedure

5.2 Practical Construction with Privacy Protection

5.2.1 Construction based on hybrid encryption schemes

A document in a procedure sometime contains sensitive information and therefore reviewers must not know some of them even if they check the application. For instance, a credit card application contains sensitive information such as name, birth-

day, address, annual income, PIN number, bank account, and so on. In this case, the card company should hide the PIN number for credit facilities and should hide the income and PIN number for a bank. To disclose a necessary part of a document for each reviewer, we improve the basic construction by applying hybrid encryption schemes[5, 8].

Applying a hybrid encryption scheme has another advantage over the basic construction. The basic construction encrypts whole plaintext $M_i (i = 1, \ldots, n)$ with a ciphertext C_{i-1} independently. Therefore, the size of a ciphertext depends on the number of reviewers and, if a plaintext M_i is large, the ciphertext size is also large. In hybrid encryption schemes, the target data encrypted by a reviewer's public key is secret keys on a symmetric key cryptosystem. Therefore the ciphertext size is not increased even if a plaintext M_i is large.

We describe an enhanced construction to disclose a part of a document in Fig. 6 and call this as ESROC$^+$. In Fig. 6, $\lambda_{\mathtt{SE}}$ is a security parameter of a symmetric key cryptosystem SE, which is a system parameter of ESROC$^+$. In addition, $\mathscr{B}_i$ denotes an access control list for a reviewer u_i, which contains an index set of blocks m_j $(j = 1, \ldots, s)$ that the reviewer u_i can read. Note that the key generation algorithm of this construction is the same as the basic construction (see in Fig. 2).

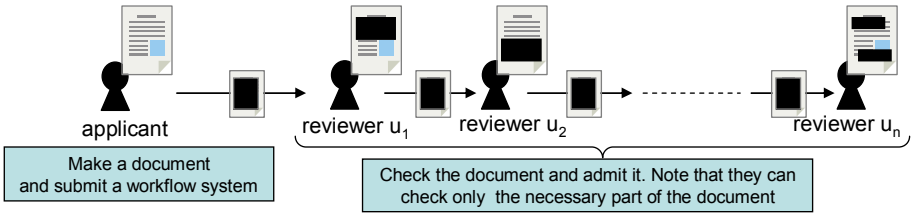

Fig. 5 Construction with Privacy Protection

$\mathtt{ESDOS}^+.\mathtt{Enc}(M; pk_1, \ldots, pk_n)$
  for $i = 1$ to s do
    $k_i \leftarrow \mathtt{SE.KeyGen}(\lambda_{\mathtt{SE}})$; $c_i \leftarrow \mathtt{SE.Enc}(m_i, k_i)$;
  end for;
  for $i = 1$ to n do
    $K_i \leftarrow Concat(hash(C), \mathscr{B}_i, \{k_j | j \in \mathscr{B}_i\})$;
      $(C = (c_1, \ldots, c_n))$
  end for;
  $C'_1 \leftarrow \mathtt{ESROC.Enc}(K, pk_1, \ldots, pk_n)$
    $(K = (K_1, \ldots, K_s))$;
  output $C_1 = (C'_1, C)$;

$\mathtt{ESDOS}^+.\mathtt{Dec}(C_i; sk_i)$
  $(C'_i, C) \leftarrow Parse(C_i)$;
  for $i = 1$ to s do $m^*_i \leftarrow$ NULL; end for;
  $(C'_{i+1}, K^*_i) \leftarrow \mathtt{ESROC.Dec}(C_i; sk_i)$;
  $(H^*, \mathscr{B}_i, \{k_j | j \in \mathscr{B}_i\}) \leftarrow Parse(K^*_i)$;
  if $H^* \neq hash(C)$ then output $\perp$;
  foreach $j \in \mathscr{B}_i$ do
    $m^*_j \leftarrow \mathtt{SE.Dec}(c_j, k_j)$;
  end foreach;
  output (C_{i+1}, M^*_i)
    $(C_{i+1} = (C'_{i+1}, C), M^*_i = (m^*_1, \ldots, m^*_s))$;

Fig. 6 Construction with Privacy Protection

5.2.2 Security

The scheme described in Fig. 6 is the same as the hybrid construction except the extra information, a ciphertext C_{i+1} and several secret keys of a symmetric key cryptosystem, are encrypted by a public key pk_i in the scheme in Fig. 6. From the result of a hybrid encryption schemes (see [5, 8, 11]), we can easily know that such extra information has no influence for the security. Therefore, from the same reason of the basic construction, if the asymmetric key cryptosystem PE and the symmetric key cryptosystem are secure enough, the proposed construction in Fig. 6 is also secure.

Note that, in the above discussion, we do not consider that an adversary regenerates the ciphertext. This is because, if an adversary who has some private key knows the whole information from ciphertexts except C_i, he/she can easily create a ciphertext C_i', which is different from C_i but generates the same resulting plaintext M_i^* as the target ciphertext C_i. In order to protect this "regeneration attack', it might be useful to attach a digital signature of the applicant with a document.

6 Conclusion

In this paper, we propose encryption schemes to enforce the order of reviewers in a workflow system. We give a basic construction and enhance it for a threshold procedure and for disclosing a necessary part of a document for each reviewer. The enhanced constructions help us to describe various procedures in a workflow system.

Conventional workflow systems are constructed with a server, which we consider a trusted one, to control the order, documents and reviewers in a procedure. Such workflow systems imply that an administrator might tamper a procedure. However, the proposed schemes strongly protect sensitive information in a document and the order of reviewers in a procedure even if the administrator is not trustworthy. Hence, the proposed schemes are especially useful for a procedure in which some reviewers belong to different corporations, e.g., a one-stop service of public institutions, or in which a corporation outsources some processes, e.g., credit facilities for a credit card application of credit card companies.

In practice, if we construct a procedure by the proposed schemes, we have to consider maintenance of the procedure, e.g., key management, changing the procedure and so on. To construct and evaluate a workflow system by using the proposed schemes is one of our future works.

References

1. "The Sarbanes-Oxley Act.", 2002.
2. Financial Services Agency, "Financial Instruments and Exchange Law", 2006. http://www.fsa.go.jp/common/diet/164/index.html (in Japanese)

3. “Private Information Protection Law”, http://www.kantei.go.jp/jp/it/privacy/houseika/hourituan/ (in Japanese)
4. XML encryption, http://www.w3.org/TR/xmlenc-core/
5. M. Abe, R. Gennaro and K. Kurosawa, “Tag-KEM/DEM: A New Framework Hybrid Encryption”, IACR Cryptology ePrint Archive: Report 2005/027, 2005. Available at http://eprint.iacr.org/2005/027
6. 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, Springer-Verlag, 1998.
7. M. Bellare, A. Boldyreva and J. Staddon, “Multi-Recipient Encryption Schemes: Security Notions and Randomness Re-Use”, Public Key Cryptography – PKC 2003, Lecture Notes in Computer Science Vol.2567 (LNCS 2567), pp.85-99, Springer–Verlag, 2003
8. R. Cramer and V. Shoup, “Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack”, SIAM Journal on Computing archive, Vol. 33 , Issue 1, Society for Industrial and Applied Mathematics Philadelphia, PA, USA, pp. 167-226, Society for Industrial and Applied Mathematics, 2004.
9. Y. Dodis and A. Ivan, “Proxy cryptography revisited”, In Proceedings of the 10th Annual Network and Distributed System Security Symposium (NDSS), February 2003.
10. Y. Hatano and K. Miyazaki, “An Encryption Method for Multiple Receivers with Different Roles”, IEICE Technical Report, ISEC2005-167, 2006. (In Japanese)
11. Y. Hatano, K. Miyazaki and Toshinobu Kaneko, “A Study on Extended Multi-Recipient Encryption: Security Notion and Constructions”, IEICE Technical Report, ISEC2007-88, 2007. (In Japanese)
12. R. Impagliazzo and M. Luby, “One-way functions are essential for complexity based cryptography”, Proceedings of the 30th Symposium on Foundations of Computer Science, pp. 230-235, 1989.
13. M. G. Reed, P. F. Syverson and D. M. Goldshlag, “Anonymous Connections and Onion Routing”, IEEE Journal on Special Areas in Communications, vol. 16, No. 4, pp. 482-494, 1998.
14. R. Rivest, “All-Or-Nothing encryption and the package transform”, Fast Software Encryption ’97, Lecture Notes on Computer Science, vol. 1267, pp. 210-218, Springer-Verlag, 1997.
15. A. Shamir, “How to share a secret”, communications of the ACM, 22(11), pp.612-613, 1979.
16. R. Zhang, G. Hanaoka, J. Shikata and Hideki Imai, “On the Security of Multiple Encryption or CCA-security+CCA-security=CCA-security”, Public Key Cryptography - PKC 2004, 7th International Workshop on Theory and Practice in Public Key Cryptography, Lecture Notes on Computer Science, vol. 2947 (LNCS 2947), pp.360-374, 2004.