# Sharing Confidential Documents Management System for Multiuser

Sung-Hwa Han
D*ept. of Information Security*
*Tonmyong University*
Busan, Republic of Korea
shhan@tu.ac.kr

2022 IEEE/ACIS 7th International Conference on Big Data, Cloud Computing, and Data Science (BCD) | 978-1-6654-6582-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/BCD54882.2022.9900560

***Abstract*— In an enterprise environment, various types of information are distributed. Information to be distributed can be distributed in text, audio, or video types, but the most basic distribution unit is a document file. Some organizations may distribute confidential information by storing it in a file. Only limited users can access these files, and unauthorized access should be blocked. In this study, we propose a confidential document file sharing technique that allows only limited users to access and check the contents. The proposed technique uses encryption technology and uses a key management system. Due to these characteristics, only pre-defined users can access. As a result of verifying the function to confirm the effectiveness of the targeted confidential document technique, it was confirmed that all the targeted functions were provided. Because this study uses encryption technology, the performance is slower than the legacy method. Additional research is needed to improve this.**

***Keywords*— *Document sharing, Confidential file, Cryptography, Registry, Document editor***

## I. INTRODUCTION

In an enterprise environment, various applications are used to share information. The information owner registers information in the web service, and other users can access this web service and review the registered information. The information type is not just text. There are images such as JPG or BMP, and multi media such as mp3 or mp4 can also be shared. If web service is used, such unstructured information can also be shared. This web service has the advantage that all users who can access the web service can access it if the information owner registers the information to be shared.

This is not appropriate from a security point of view. Important information should be accessible only to limited users. For this, identification or authentication is enforced on the web service so that only authorized users can access. This security function enforced web service has already become popular and is being applied to many web based services.

But this alone is not enough. After the authorized user has access to important information, the information can be downloaded to the user system. In this case, the owner of important information is the user who downloaded it. The download user can own the information only to himself, but he can share it with other users if he wants. The user who received the share may be a user that the information owner does not allow for the first time. From here, the value of important information drops. Downloaded information can no longer be important information [1].

This security environment is not appropriate in an enterprise environment. When necessary, only very limited users should be able to access information generated in an enterprise environment. Even if the limited user has access to the information, another copy should not be created. Also, important information should not be transformed into another form and shared with other users.

In this study, we propose an enforced confidential document sharing system as the first process to satisfy this security requirement. The proposed system enforces password techinque so that only limited users in the enterprise environment can access. Also, limited users cannot create copies even if they access confidential documents and cannot transform them into other forms. Because this security function is enforced, the information owner can keep the information value.

## II. RELATED STUDY

### A. Enterprise environment for document usage

In an enterprise environment, a variety of information is handled. There is information that can be opened externally, and there is information that cannot be opened externally and can be opened internally. In particular, confidential information cannot be opened internally, and only limited users can access it. The term of information is conceptual. It has a value without a form. Since this information cannot be used by itself, it must be expressed in a usable unit, and the expressed information must be stored so that it can be used at other times [2].

The storage method of information is different depending on the purpose, usage method, and volume of the information. When using only a single user, the file storage method is usually used. If the storage capacity is large, it can be stored in the database. When there are many information users, web service can be used, and for information with frequent changes, configuration system can be used. Users use various applications so that the stored information can be used when needed, depending on the type of storage. An application capable of handling stored information can be unique, but usually a variety of applications can handle it. In particular, in an enterprise environment where user convenience must be supported, universal methods such as office files such as word, excel, ppt, and pdf are used. Since the mechanism of this information

method is well known, various applications support this method [3].

Information itself guarantees confidence and integrity. However, when this information is stored for use in an enterprise environment, many users access it using many applications [4].

### *B. Security threats about confidential document sharing*

The information used in an enterprise environment is not always at the same level. While there is open-able information, there is also confidential information. In general, confidential information can only be accessed and modified by internal limited users because of confidentiality and integrity. However, exceptions always occur [5].

What is of concern is when and where confidential information is used. Confidential information is not always used internally. If necessary, it can be exported to external. In addition, only internal users do not access, and when approved, external users can also access. Information is not physically unique. When a user who has accessed the information accesses the information, the information is copied [6]. This is a function provided by the application for application to display saved information to the user. This application usually provides a 'Save as' function, which allows you to create a copy of the information. Once a copy of information is created, the confidentiality of the information is no longer guaranteed [7].

Fig. 1 shows a representative case of information leakage. The owner uploads it to the web server. The web server stores confidential information uploaded by the owner in the server. The internal user accesses confidential information registered in the web server and checks its contents. And a copy of confidential information is stored in the user device. Ownership of the copy of information is also held by the internal user who created the copy. Since the internal user has information ownership, it can be modified to make another copy or distribute it to other users. If an internal user distributes confidential information to other users to external users, confidential information is no longer confidential. It changes to normal information. If such a threat occurs in an enterprise environment, the organization is damaged by confidential information leakage [8].

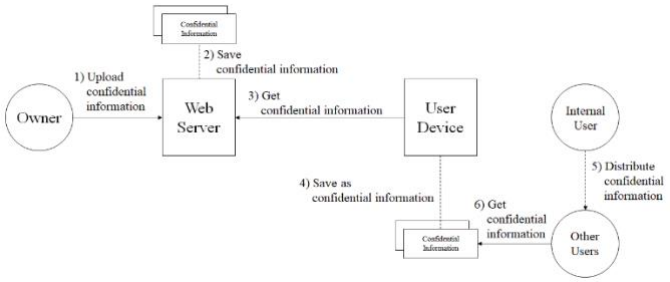


Fig. 1. Confidential information leakage process

## III. Security Requirement for Current Enterprise Environment

### *A. Security environment analysis*

Confidential documents contain important information, which is dynamic. As time passes, the important level of confidential documents changes. In an enterprise environment, confidential documents can be continuously created and modified. In addition, confidential documents whose use has expired are terminated so that they are not misused.

Confidential documents can be accessed by various users. Users can share it internally, and when necessary, it can be transferred to an extetnal user. Such sharing or transfering is authorized. Therefore, when the owner of the confidential document authorizes it, it can be shared or transferred to other users.

However, users' mistakes occur even when handling confidential documents. Or, even in an enterprise environment with confidential documents, unauthorized access also occurs. Users with low security level can access confidential documents. Alternatively, a malicious attacker could access confidential documents from a remote location through an internal service. Both cases are unauthorized access, and if a separate protection technique is not applied to the confidential document, important information stored in the confidential document may be leaked or modified.

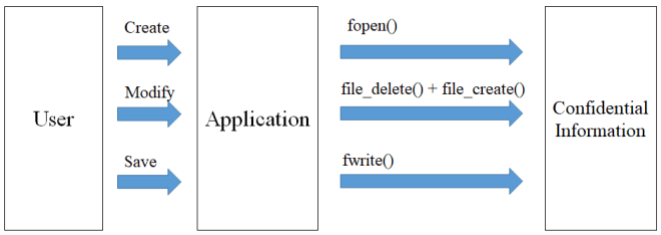


Fig. 2. Document handling process by applicaion

This is caused by the handling mechanism of the application used to access the document. The mechanism for the user to check the information stored in the document is shown in Fig. 2. Many system calls are executed when the user selects and opens a document after running an application that can handle documents. When a user creates a document for the first time, a temp file is created. When a user creates or saves a document, the file-open systemcall is executed in write mode. When reading the stored document, it is executed in append mode. If the document is opened repeatedly by the user, the post open is opened in read-only mode. Although it is different for each operating system, when modifying and saving a document, delete systemcall and create systemcall are usually executed in order. As such, the executed system calls are all different according to the document handling function selected by the user.

### *B. Security requirement for document sharing*

In an enterprise environment, confidential documents can only be used. Confidential documents can be used both internally and externally [9]. However, confidential documents should be used safely regardless of the location and time of use [10]. Security functions for protecting confidential documents must not interfere with the enterprise environment [11]. It should support the achievement of the enterprise object. Therefore, it must support various enterprise business processes [12].

Table 1 shows the sequence requirements for confidential document using in an enterprise environment. Each security

requirement is to respond to security threats. When a security requirement is met in an enterprise environment, a security threat may not occur or may be rejected. Conversely, if this security requirement is not met, a security threat occurs [13].

TABLE I. SMART FACTORY LEVEL AND MAIN FUNCTION

| Security Requirement | Description |
|---|---|
| Identification and Authentication | · Access of authorized users is allowed, but access of unauthorized users must be denied.<br>· For this, the access control mechanism must be able to identify the user and authenticate. |
| Authorization | · It should be possible to set detailed privacy for a user or user group.<br>· The document should be able to monitor the user's create, read, write, modify, and delete behavior, and control each behavior. |
| Confidencity | · Confidential document should be able to confirm its contents only to limited users.<br>· Even if access is not possible, confidential documents can be accessed if the file is copied from the external side.<br>· In order to deny confidential document access by user file copy, confidential documents must be encrypted to ensure confidenity. |
| Clip-board control | · You can export contents to external by copying them to clip-board.<br>· Therefore, users who access confidential documents should be able to block clip-board use. |
| Integrity | · A read-only file can be modified by unauthorized behavior.<br>· Such unauthorized modification should be denied immediately. |
| Sharing | · Even if it is a confidential document, only the owner cannot use it.<br>· Many users should be able to access it.<br>· However, access control should be enforced for every many users. |

Security requirement does not implement security function for each requirement item. Therefore, it is necessary to design the security service architecture after considering many security requirements.

## IV. CONFIDENTIAL DOCUMENT SHARING SYSTEM

### A. System architecture

This study proposes a confidential documents management system for multiuser to meet the security requirements presented in Table 1. The proposed document management system can identify and authenticate users who want to access confidential documents. An authorized user verified by applying identification and authentication can access confidential documents. However, the result of the user identification and authentication process is the user's privacy. This access control mechanism controls user access to documents. Through this, all document handling executed by the user is monitored. When user document handling (e.g. create, read, modify, save, delete) behavior is allowed in the privilege, user behavior is allowed. Conversely, user behavior is denied when privilege does not allow it.

The overall system architecture of the confidential documents management system for multiuser proposed in this study is shown in Fig. 3.

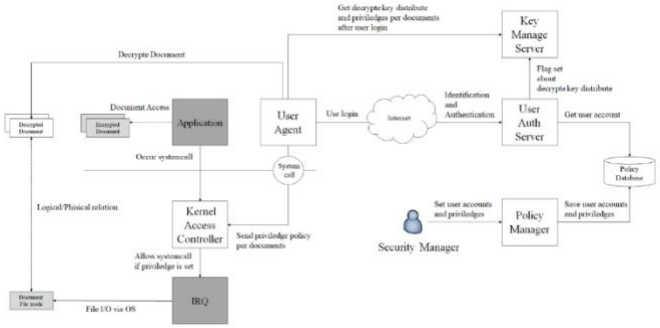


Fig. 3. Confidential documents management system for multiuser

This study proposes a confidential documents management system for multiuser to meet the security requirements presented in Table 1. The proposed document management system can identify and authenticate users who want to access confidential documents. An authorized user verified by applying identification and authentication can access confidential documents. However, the result of the user identification and authentication process is the user's privacy. This access control mechanism controls user access to documents. Through this, all document handling executed by the user is monitored. When user document handling (e.g. create, read, modify, save, delete) behavior is allowed in the privilege, user behavior is allowed. Conversely, user behavior is denied when privilege does not allow it.

The overall system architecture of the confidential documents management system for multiuser proposed in this study is shown in Fig.3. The confidential documents management system for multiuser proposed in this study consists of 6 components. User Auth Server identifies and authenticates users. Enable user document key sharing only when the user is authenticated. Policy Manager defines users who can access confidential documents and sets user privacy. If necessary, Security Manager can audit encrypted documents, decrypted documents, and authenticated users through Policy Manager. The user account and privilege set by the Security Manager are stored in the Policy Database. Key Management Server is a server that stores an encryption key that can decrypt encrypted documents. This encryption key is transmitted only to the authenticated user.

The user agent operating on the user device is responsible for the role of the login interface of the user who wants to access the encrypted document, and the role of decrypting the encrypted document and converting it into a decrypted document. When a user is authenticated, the file accessed by the user is decrypted. The User Agent is also responsible for delivering the privilege policy set by the Security Manager to the Kernel Access Controller. Kernel Access Controller monitors the application's system call used when a user accesses a Decyrpted Document. If the type of system call generated by user behavior violates the privacy policy, deny it. Conversely, if allowed by the privilege policy, ignore it and deliver it to the IRQ to handle file I/O by the OS.

### B. Crypto key management

In order to use confidential documents in which important information is stored in an enterprise environment, the

confidential documents are encrypted. And in order to access confidential documents, the encrypted document must be decrypted and converted into a plain document.

When a confidential document is encrypted with a single encryption key, all users with the encryption key can access it. However, here, the encryption key may be transmitted to other users due to the user's negligence or intention. When this situation occurs, the important information stored in the confidential document is no longer important. Only normal infomration is saved. If the information stored in this document is used by a malicious attacker, it can leankage the company's importtant information and cause damage to the company.

Confidential documents should be accessible only to limited users. Also, even if the encryption key of a limited user is leaked to the outside by negligence or intention, the damage should be minimized. If a single encryption key is used, all encryption keys must be changed. If the encryption key of a specific user is exposed to the outside, the encryption key must be disabled immediately. Therefore, when encrypting confidential documents, the encryption key must be enforced separately for each user.

If the confidential document is encrypted with the encryption key each user has, there is an inconvenience of having to make a copy of each. In this way, the capacity increases and the number of objects to be managed increases. Confidential documents to be protected should be minimized. It is easy to manage only one confidential document. To this end, in this study, the confidential document encryption key management system is applied.

The key management mechanism proposed in this study enforces the double encryption key. Confidential documents storing important information to be protected first are encrypted with a document crypto key. This document crypto key is encrypted again with the user key. This user key is stored in the key management server. User keys are only passed to authenticated users. It is stored encrypted with the User Password in the Key Management Server.

If this encryption key management system is applied, there is an advantage that the encryption key can be protected even if the key management server where the encryption key is stored is attacked by a malicious attacker. Also, even if a specific private key is opened, only the user key needs to be disabled, so key management is easy. In addition, since the document crypto key finally used for document decryption is encrypted with the user key, there is an advantage that only one confidential document can be sufficiently protected. Therefore, in this study, the double encryption key management system was enforced.

## V. IMPLEMENT

### A. Verification environment and items

The confidential documents management system for multiuser proposed in this study must be able to accurately provide the targeted security function. Therefore, it is necessary to verify all security functions proposed in this study.

The evironment to verify the target security function of the confidential documents management system for multiuser proposed in this study is divided into hardware and software. The hardware environment in which the User Auth Server and User Agent operate was implemented in an i5-8750 CPU with 8Gbyte Memory and 256Gbyte SSD. The software was implemented in Redhat Enterprise Linux 8.4 environment. The software in which the user agent operates was also verified in the Redhat Enterprise Linux 8.4 environment.

Implemented according to the architecture of confidential documents management system for multiuser proposed in this study.

The security function targeted by the confidential documents management system for multiuser proposed in this study is diverse. Only when all functions operate normally, the target confidential documents management system for multiuser can be implemented. Therefore, in this study, verify items as shown in Table 2 were defined.

| Item ID | Description |
|---|---|
| Func_01 | · Check that the authorized user logs in normally.<br>· When a user logs in using ID/PW, check whether the user key is delivered to the user agent. |
| Func_02 | · When the user agent receives the user key, it checks encrypted document is decrypted using the user key.<br>· If the user agent decrypts the encrypted document normally, the contents can be checked by accessing the plain document using the application. |
| Func_03 | · Check whether unauthorized users access encrypted documents in Func_01 state or not.<br>· When an unauthorized user accesses an encrypted document, since this document is encrypted, the unauthorized user cannot check the contents. |
| Func_04 | · In Func_02 state, when a user executes unauthorized behavior, the Kernel Access Controller detects it accurately and checks whether the systemcall is denied.<br>· If a user who is allowed only read-only privilege does save-as, the system call corresponding to this behavior is denied. |

### B. Verification result

As a result of verifying Func_01, it was confirmed that the user with account information registered correctly logged in. Conversely, it was confirmed that users without account information could not log in even if they entered ID/PW. Therefore, it can be determined that Func_01 operates normally.

Func_02 was verified. As a result, it was confirmed that the User Auth Server delivered the User Key stored in the Policy Database to the User Agent when the user logged in. In addition, when the user agent received the user key, it was confirmed that it was used to decrypt the encrypted document. Finally, a plain document was created. Therefore, it can be determined that Func_02 operates normally.

Func_03 was verified. It was confirmed that the contents could not be checked when an unauthorized user accessed the encrypted document.Even if an unauthorized user executes the application and accesses the encrypted document, it was confirmed that an error occurred and the document information could not be reviewed. Therefore, it can be determined that Func_03 operates normally.

Func_04 was verified. If the user performs unauthorized behavior, the Kernel Access Controller checks this to determine

whether the execution is denied. As a result, it was confirmed that Kernel Access Controller blocks system calls generated by unauthorized behavior.

## VI. Conclusion

In the enterprise environment, various information is utilized, and document is used as a representative information storage method. Various information may be stored in this document, which may include confidential information. This confidential information can only be used in various ways. However, there are many security threats in such confidential information.

In this study, to deny such security threat, a confidential documents management system for multiuser was proposed. To verify effectiveness, function and performance were checked. As a result, it has been confirmed that the targeted security function is accurately provided and the performance is sufficient. Therefore, it can be judged that the confidential documents management system for multiuser proposed in this study is sufficiently effective.

However, this study assumes a method that uses only ID/PW, and there is a limitation in not applying various authentication methods used in an enterprise environment, so additional research is needed.

## References


[1] van Renesse, R. L.: Paper based document security-a review. In European Conference on Security and Detection. 1997. ECOS 97, IEC, pp.75-80, 1997.

[2] Kudo, M. and Hada, S.: XML document security based on provisional authorization. In Proceedings of the 7th ACM conference on Computer and communications security, pp.87-96, 2000.

[3] Müller, J., Ising, F., Mainka, C., Mladenov, V., Schinzel, S. and Schwenk, J.: Office document security and privacy. In 14th USENIX Workshop on Offensive Technologies (WOOT 20), 2020.

[4] Bhatti, R., Bertino, E., Ghafoor, A. and Joshi, J. B.: XML-based specification for Web services document security. Computer, vol.37, no.4, pp.41-49, 2004.

[5] Spannenburg, S.: Developments in digital document security. In Optical Security and Counterfeit Deterrence Techniques III, SPIE, 3973, pp.88-98, 2000.

[6] Schell, K. J.: History of document security. In The History of Information Security, Elsevier Science BV, pp.197-241, 2007.

[7] Salter, M. B.: International cooperation on travel document security in the developed world. In Global mobility regimes, Palgrave Macmillan, New York, pp.115-129, 2011.

[8] van Renesse, R. L.: Ordering the order: a survey of optical document security features. In Practical Holography IX, SPIE, Vol. 2406, pp.268-275, 1995.

[9] Khan, R. A. and Lone, S. A.: A comprehensive study of document security system, open issues and challenges. Multimedia Tools and Applications, vol.80, no.5, pp.7039-7061, 2021.

[10] Deshmukh, M. P. S. and Pande, M. P.: A study of electronic document security. Journal of Computer Science and Information Technology, vol.3, no.1, pp.111-117, 2014.

[11] Triand, B., Effendi, S., Puspasari, R., Rahmad, I. F. and Ekadiansyah, E.: Digital Document Security on Legalize Higher Education Diplomas with Digital Signature and SHA-1 Algorithm. In 2019 7th International Conference on Cyber and IT Service Management (CITSM), IEEE, vol.7, pp.1-5, 2019.

[12] Lee, R. A.: Micromanufacturing for document security: Optically variable devices. In Micromanufacturing and Nanotechnology, Springer, Berlin, Heidelberg, pp.131-169, 2006.

[13] Frankel, S. and Krishnan, S.: IP security (ipsec) and internet key exchange (ike) document roadmap. Request for Comments, 6071, 2011.