

# Design and Implementation of a Document Encryption Convergence Program Selecting Encryption Methods, and Integrating the Program into the Existing Office System

Hong-Jin Ryu<sup>1</sup> <sup>a</sup> and Samuel Sangkon Lee<sup>2</sup>  <sup>b</sup>

<sup>1</sup>*Dept. of Computer Engineering, Jeonbuk National University, Jeonju, Chonbuk, South Korea*

<sup>2</sup>*Dept. of Computer Science and Engineering, Jeonju University, Jeonju, Chonbuk, South Korea*

**Keywords:** Ancient Cryptography, Modern Cryptography, Shift Encryption, Poly-Alphabetic Substitution, Transposition Cipher, Nihilist Encryption, DES and AES Encryption, MVVM.

**Abstract:** The article explores the limitations of current encryption methods and proposes a new encryption system that combines ancient and modern cryptography through software engineering. The article also discusses the history of cryptography, from simple substitution encryption methods to more complex mathematical algorithms. The proposed encryption system fuses several ancient ciphers with AES-256 encryption, creating a more secure and stronger password that is difficult to decrypt. This paper suggests the usefulness and reliability of cryptographic algorithms, such as virtual currency and blockchain fields, and could be used easily by the general public in electronic devices.

## 1 INTRODUCTION


Cryptography has played an important role in safely storing and transmitting information. Today, many researchers are trying to enhance existing encryption algorithms more sophisticated and powerful. These encryption techniques also contributed to the creation of modern cryptosystems. Special institutions have used encryption for storage for information protection or transmission of information between institutions. However, with the spread of Internet technology, encryption technology has begun to be required for information transmission between individuals. However, programs that allow individuals to easily generate their own ciphertext through encryption algorithms have not been common until now.


Encryption is a security device that makes it unreadable by anyone other than the person concerned. It is a kind of information hiding using a computer algorithm. This aims to prevent unauthorized third parties from obtaining or viewing one's information. However, the public usually does not know how to securely encrypt their data or which encryption methods are safe to use. Therefore, it is not

easy to encrypt documents for personal purposes. Most applications are at the level of preventing others from easily opening them by putting a password on them. For this reason, even intermediate level users often store important documents without encryption. Information protection mainly consists of three types: preventing anyone from entering, preventing the information inside from leaving, and making the information useless even if it flows out. This paper intends to strengthen information protection through the third case.

In this research, we deal with the direction that a third party can never decrypt when a user's data is transmitted. We are planning to design and implement a new encryption system that combines ancient and modern cryptography through software engineering. It aims to make such an encryption program easy to use.

Advances in encryption technology mean that passwords may be cracked by third parties. Therefore, if the vulnerability of the currently used password is found, a better encryption method has to be developed. Through such efforts, encryption technology has been continuously developed from

<sup>a</sup>  <https://orcid.org/0000-0003-2042-8330>

<sup>b</sup>  <https://orcid.org/0000-0001-9965-8387>

ancient times to modern times. At first, it started with a substitution encryption method that simply replaces letters with other letters. Over time, a method of changing the order and position of letters at the same time and a transposition cipher that moved them to a different position were developed. Using the incredible power of modern computers, ancient passwords were easy to crack. Therefore, in order to deal with this, cryptography using mathematical methods was created. As computer performance increases, not only encryption methods become more difficult, but also decryption technologies. The developed encryption technology has evolved and is highly dependent on the development of computer performance. As the computer processes calculations that humans can do at a very high speed, the encryption technology using it is great, but as the performance of the computer improves, the current encryption technology cannot help but become weak. This can be seen as a limitation of the generation of cryptographic algorithms that depend on computer performance. The more difficult the algorithm used for encryption, the higher the computer performance required. To decipher it, a computer with a higher specification is required.

Although most current encryption methods tend to depend on mathematical encryption algorithms, most of the algorithms are publicly available. The fact that the encryption method is open means that anyone can create an algorithm to decrypt it if they want. Of course, even if it is a public algorithm, stability is proven to the extent that it cannot be easily deciphered. That's why the algorithms are still being used. However, even proven algorithms may not be stable anymore after the invention of future computers. One of the ways to solve this problem is to increase the length of the encryption key. However, even if an encryption algorithm with a key length of 128 bits is used, decryption may be possible if calculation is performed using a quantum computer. Of course, quantum computers are not yet widespread among ordinary users. However, it may be obvious that as computer performance develops, there is some risk that existing passwords can be cracked (Paar, and Pelzl, 2010).

Moreover, the current issue is in the case of an authentication method using biometric recognition among encryption technologies. There is a weakness that spoof authentication can succeed if a sufficient number of attackers unite and attempt biometric authentication (Richard, 2001). Therefore, in the case of encryption using biometric authentication, it is usually appropriate to mix other encryption methods. In other words, so as to compensate for the weakness

of the currently used encryption algorithm, several types of encryption methods must be mixed and used. This idea is the main concept of this paper.

In this research, the existing algorithm is utilized, but we will not rely solely on the existing algorithm. We will focus on technologies that make it less likely to be decrypted in the future. To create this, we need to mix the classical and modern encryption methods of human heritage. The best way to verify the performance of an algorithm is to release it and test it. Users can choose any of these proven algorithms to mix ancient and modern cryptography. Thus, it can be stored with multiple encryptions. Of course, to decrypt, the user must know the encryption method that combines up to 10 selected encryption methods in reverse order. The values of each key must also be known. There is an advantage in that a user performs multiple encryptions in an encryption method dependent on existing algorithm. Therefore, the encryption strength is greatly increased compared to the existing methods. Each encryption method is described in the next section.

## 2 THEORETICAL BACKGROUND

### 2.1 Ancient Ciphers

In the shift encryption method belonging to ancient cryptography, cipher text is generated by shifting 26 letters of the English alphabet to the right or left as much as the shift value (number) that becomes the key. It seems relatively simple. 26 letters of the English alphabet are shifted by 2 to the right. To use this method, keys must be applied sequentially throughout the input statement to complete. In this paper, the shift encryption method, and it was expanded to include 11,172 complete Korean characters, special characters (44), and a total of 96 characters (44+26+26) of English uppercase and lowercase letters (26+26). Related books published on the market use only English letters to explain the implementation principle (Paar, and Pelzl, 2010). In contrast, in this paper, the public can take a look at the encryption and decryption process, including Korean. Special characters, uppercase/lowercase English characters, and complete Korean characters were converted into numeric values of ASCII code and Unicode. After that, spaces between numeric values were removed, and shift encryption was applied. After that, the substitution operation was performed as much as the value corresponding to the

blank between the numeric values removed before encryption. It was implemented so that only the characters within the range that can be input are output normally. When entering a positive integer as a key value, it is shifted to the right. When inputting a negative integer, it is designed to be shifted to the left.

According to scholars, the polyalphabetic substitution method is called a 'multi-character cipher.' A typical example is the Vigenere cipher. This encryption method also belongs to the classical encryption method like the previous studies described above. In general, 26 English alphabets are filled in horizontally and vertically ( $26 \times 26$ ) size blocks so that one letter per column and row is moved. Then, the document is encrypted by configuring the key used for encryption and the input text to be encrypted in horizontal and vertical coordinates, respectively (Lee, Yeom, Song, and Kim, 2005).

The encryption of the polyalphabetic cipher method is generally made by adding 11,172 complete Korean characters and 96 ( $=44+52$ ) characters such as special characters and English uppercase/lowercase characters, like the shift cipher above, based on the 26 characters of the English alphabet. Each character in the input statement is converted to Unicode value, and code values not used within the program are excluded and arranged in continuous sequence of numbers. The same operation was applied to the key values as well. The value is added to the calculated input statement, and it is restored as a character. This restored character becomes the ciphertext. In summary, in this paper, the principle of the existing poly-alphabetic cipher text produced only with English letters is used as it is. However, additionally, an algorithm for generating a password was implemented by extending it to Korean, English, and special characters. In this way, when encrypting the text in a document written by the people, a program was developed that can encrypt all characters normally.

Polyalphabetic cipher is a standard transposition cipher belonging to the classical cipher, and can be classified as a block cipher using blocks (Seo, 2017). It is created by substituting the input text to be encrypted into a block with as many columns as the input key value, which must be numbers, in row units and then outputting it in column units.

In the program designed in this paper, standard transposition cipher is implemented and is operated in the same way as mentioned above. The standard transposition cipher, like other ciphers, is designed so that it can be applied without error to documents containing not only English but also complete Korean

and special characters. A key transposition cipher is similar to a standard transposition cipher in that it uses a block cipher. However, compared to the existing standard transposition encryption where the input key value was limited to numbers, the program designed in this paper is different in that it allows text key values as well. It provides improved encryption by further increasing the interpretation strength. Key-type transposition encryption has the advantage that numbers, letters, or special characters can also be used as key values. The encryption method creates a block with as many columns as the length of the input key value (characters) and substitutes the input text like a standard transposition cipher. After that, the key value (character) is replaced with the same number as the alphabetical order, and the ciphertext is completed by outputting the column with the smallest number.

In the key transposition (key type transposition) cipher, a block with as many columns as the length of the key value is created, and then the input text to be encrypted is substituted. After that, the key values are output in column units in alphabetical order to complete the ciphertext.

Among classical ciphers, the Nihilist cipher is a type of transposition cipher and is a block cipher. It uses an encryption method similar to a key-type transposition cipher. The difference is that the ciphertext is completed by applying the key value by column unit and then applying the key value by row unit (according to alphabetical order) and outputting it. Therefore, ciphertext with higher strength than standard transposition cipher or key-type transposition cipher is generated. So far, the implementation principles of five ancient ciphers (Shift, PS; Polyalphabetic Substitution, STC; Standard Transposition Cipher, KTC; Key Transposition Cipher, and Nihilist) have been described. The next section describes the design of modern cryptography.

## 2.2 Modern Cryptography

In 1975, the National Bureau of Standards of the United States established a kind of block (Seo, 2017) encryption called DES (Data Encryption Standard) as a standard for data encryption. It is a national standard set by the US NBS (National Bureau of Standards, now NIST). This technology is a symmetric-key encryption method using a 56-bit key. We have been using it for a while without any problems. However, if the length of the key is too short and a backdoor is included, it can be decrypted in a special way, which raises several problems. Therefore, it was not long

before DES was found to be weak and was replaced by the modern AES (Advanced Encryption Standard) encryption method.

AES-256 is the original name of the Rijndael cipher. This is a type of modern cryptography. As mentioned above, AES-256, the national standard encryption method of the United States, was designed to secure the vulnerability of DES, which was used as the existing national standard encryption method. It is known to be the strongest among currently used encryption methods (Nakov, 2018). AES is a 128-bit block encryption method newly developed to solve the problems of DES encryption technology. AES technology also uses a symmetric key method by using the same key in the existing encryption and decryption processes. It differs from DES in that blocks of 128 bits can be processed with key lengths of 128, 196, or 256 bits.

Wi-Fi, which is the short-distance wireless communication technology that we use the most, also uses AES as one of the encryption methods. In the case of Bluetooth, the SAFER+ encryption technology was used up to the 3.0 standard, and the AES standard is used from Bluetooth 4.0. In this paper, DES and AES, which are modern encryption methods, are implemented. For more effective and accurate encryption program development and convenient implementation, System.Security.Cryptography of the C# library was used. This library helps to perform several tasks such as encoding/decoding hashes, generating random numbers, and authenticating messages. It also provides a way to easily implement encryption methods including data. In paper, AES-256 was applied and a key value of 256 bits.

### 3 ENCRYPTION DESIGN

The system uses a mix of classical and modern ciphers to encrypt documents and store the ciphertext. Conversely, it was designed with the goal of decrypting the stored cipher text and restoring it to the original document. The following Figure 1 shows the overall system structure diagram. The function requirements are as follows. The function of loading a document enables a text file to be loaded and displayed on the screen. After that, when selecting an encryption method, up to 10 encryption methods used are allowed to be duplicated, so that mixed encryption methods can be selected. This method is a convergence encryption method that allows a total of seven types of ancient and modern passwords to be duplicated up to 10 times. This paper is differentiated

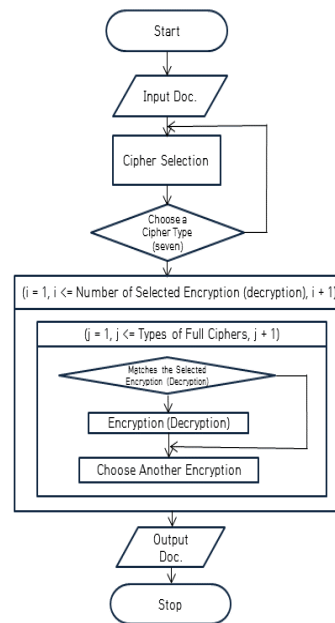


Figure 1: Overall Control.

from other papers in that it can combine various types of encryptions in various forms.

When entering a key value to be used for encryption, it is necessary to be able to input each key value to be used according to the type of encryption. Here, the type of encryption algorithm or key length is an important factor in determining the security level of the system. In particular, KISA (Korea Internet & Security Agency) recommends that the length of the private key must be 160 bits or more. It is also recommended that the expiration date be up to two years. The proposed method of this paper contains seven encryption methods (Shift, PS (Polyalphabetic Substitution), STC (Standard Transposition Cipher), KTC (Key Transposition Cipher), Nihilist, DES, and AES-256). In addition, the above seven can be applied in combination, and the order can be changed. Depending on the order in which these applications are applied, the resistance to decryption of ciphertext varies greatly. Functions such as ciphertext output (encrypted text is displayed on the screen), ciphertext storage (ciphertext can be saved as a text file), and encrypted document decryption (the original text is restored by decrypting the ciphertext) are all included.

From the standpoint of software engineering, the quality requirements of this program are as follows. Input and output (retrieving and saving input text and cipher text should work normally, and there should be no data loss). It is designed to make decryption impossible if the original text is modified even

slightly. Figure 1 explained the flow chart of the entire encryption/ decryption process of this program. Next, the use case actor specification and outline specification are expounded. As shown in Table 1, this program has one actor, 'user.'

Table 1: Specification of Use-case "Actor".

Actor	Description
User	<ul style="list-style-type: none"> <li>The user is the main agent who uses the program.</li> <li>The user can use the program to open and save text.</li> <li>The user can use the program to select or add document encryption methods.</li> <li>The user can use the program to encrypt documents.</li> </ul>

From the user's point of view, use-cases are used to easily explain the scope and function of the system, which is called 'usage pattern' in Korean. A use case is literally a "purpose that is used" to prevent chaos in which two things are used at the same time. A use case is a multi-purpose system created by gathering the uses of the system. In this way, if use cases are collected and connected to the system, users can easily understand the development process. The use cases of this program include 'document open', 'password selection/adding', 'key registration', 'document encryption', and 'document saving (Save)' and 'Document Decryption (Decryption)' are added.

Sequence diagrams were created using Web Sequence Diagrams<sup>3</sup>. Here, the sequence diagram describes the sequence of message exchange between objects of interaction. Presents a Unified Modelling Language (UML) diagram. In particular, UML is used for a model that conceptually and physically expresses a system with vocabulary and rules for creating software. It helps to solve problems of system structure, communication within the project team, and reuse of software structure. The design details used in the program of this paper are shown. The design details used in the program of this paper are shown. As exhibited in Figure 2, it was designed as a sequence diagram.

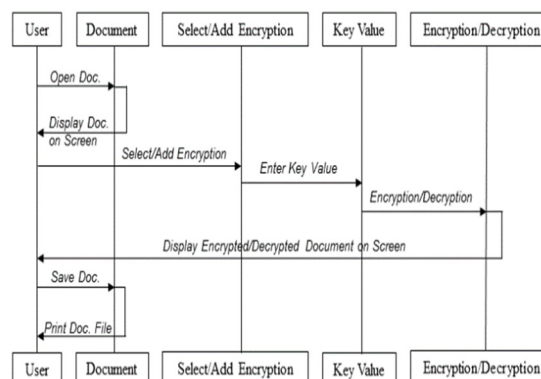


Figure 2: Sequence Diagram.

## 4 IMPLEMENTATIONS

### 4.1 User Interface

The screen design was composed of a user-friendly design. The menu is placed at the top, and the toolbar below it is used to quickly access the menu. In the window showing the result of the program, a window for outputting the input text and a window for outputting the converted ciphertext appear in pairs.

### 4.2 Structural Design

#### 4.2.1 Architecture Model

An important design pattern in program development was developed using the MVVM (Model-View-ViewModel) model as shown in Figure 3 below. This model is different from the existing MVC (Model-View-Controller) or MVP (Model-View-Presenter) patterns. As a software architecture pattern, the development of a graphical user interface (view) implemented in a markup language or GUI code is separated from business logic or back-end logic(model). This ensures that views are not tied to any particular model platform. It is efficient for UI accessibility or interpretation of program source code. In the picture, INPUT means user's input or event. The solid line means information exchange in a general form, while the dotted line means that information exchange is possible, but it is not recommended for reasons of dependency between models and security. In addition, since \* means many and 1 means one, MVC on the left means a many-to-one relationship. MVP stands for one-to-one

<sup>3</sup> <https://www.websequencediagrams.com/>

relationship. Finally, MVVM stands for one-to-many relationship.

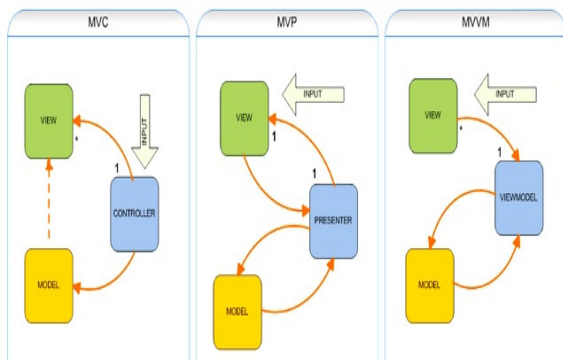


Figure 3: Design Patterns.

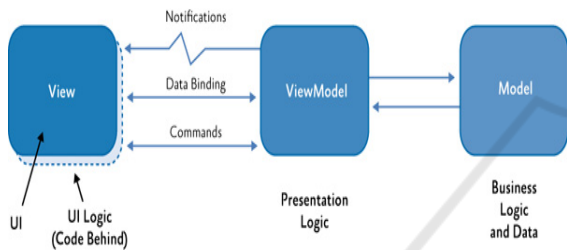


Figure 4: Example of MVVM.

In the existing MVC pattern, the controller directly accesses the view and model and controls them. Therefore, the model was able to access the view indirectly. In the MVP pattern, the view and model are accessed through the Presenter. However, the MVVM pattern adopted in this program is designed with three structures: View, Model, and ViewModel, as shown in Figure 4. Design is designed in the view, data is processed in the model, and data is bound using model information in the view model. The view is so controlled. When designing the view on the left side of the figure, the UI (User Interface) composed of XAML and the code composed of an easy-to-develop language are designed on the code behind to configure the User Interface Logic. The notification in the middle of the picture operates the view according to the instruction of the view model. When this operation is performed, information is exchanged in data binding between the view and the view model. At the same time, commands are executed. The view model corresponding to the presentation logic controls the overall parts displayed in the view. The model corresponding to Business Logic and Data on the right side of the figure plays a role in managing information storage, modification, and deletion.

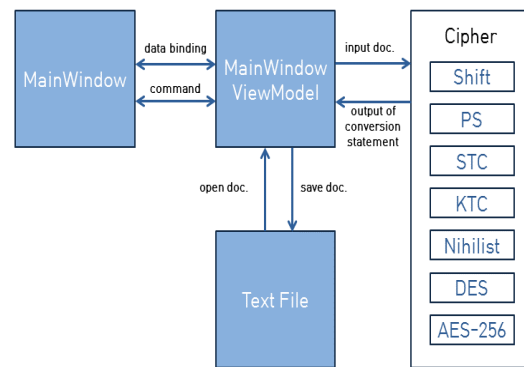


Figure 5: System Architecture.

Thus, the MVVM pattern is more modern than the previous two design patterns. Efficiency can also be achieved with a clear separation between the domain logic and the presentation layer (separating the model from the view obtained from the user's point of view). MVVM provides clean separation of code and is therefore maintainable. Separation of core logic parts into external and internal dependencies, in other words, it is easy to test unit module programs for core logic. Reusing code, replacing code, or adding code to the right place in the architecture is appropriate. Abstraction can also be achieved with code-behind. As described above, our program designed more efficient program by using the MVVM model.

Figure 5 above shows the structure of this system. First, we created the Main Window to be used as the view. The source code was written using WPF's XAML. I created a Main Window View Model for the main window, which will be the view model. By binding data with the main window, when data changes occur in the main window view model, it is implemented so that it can be immediately reflected in the view. In addition, it is implemented so that each function works well so that user commands are transmitted between the view and the view model by using a command. In the main window view model, a text file corresponding to the model can be loaded or saved. The Cipher on the right side of Figure 5 can be accessed.

#### 4.2.2 String Operations

This program uses regular expressions and string internal operations to process strings using XAML link in Section 4.2.1. In the case of the transposition cipher, the input statement does not require any intra-string operations. However, key values require internal operation. In particular, the substitution is

required by internal string operation for both input text and key values.

In the shift encryption method and the standard transposition cipher (STC) method, the input text can be in any form. However, since only numbers are allowed as key values, the input key value must be converted to an integer (int) type for calculation. In addition, the key value of the shift cipher must receive a plus (+)/minus (-) value indicating two directions (right or left). After the input text is converted into an ASCII code or a Unicode value, internal operations are performed. This internal operation is implemented when the cipher text is output after the encryption process is finished or, conversely, numbers are replaced with characters and output. Moreover, the same process applies to the encryption process as well as the decryption process.

### 4.2.3 Implementation and Function

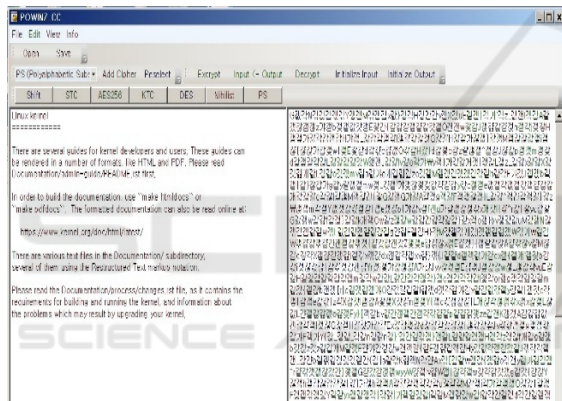


Figure 6: Encryption of Text Files.

This section explains the menu composition in detail and at the same time helps users understand each encryption. When you press the combo box, a menu composed of seven items such as Shift, PS, STC, KTC, Nihilist, DES, and AES-256 appears. Prior to selecting a password, the Open/Save toolbar menu is empty. This program shows an example of choosing four passwords. This is the screen where the currently opened document is encrypted in the order of Shift, PS, STC, and KTC. If you select a specific encryption method from the combo box and select the Select button, a button with the encryption written on it is added. Up to 10 combinations of methods can be selected. In the future, it will be upgraded to enable changes (add/modify/delete) of the selection method. The selected encryption is calculated in the same reverse order in the encryption or decryption process. If you click the Reselect button, all currently selected and added encryption methods are cleared. Also, the

toolbar that outputs the password selected so far is initialized at the same time. The shift encryption method and the STC encryption method allow only numbers to be entered as key values. The encryption key is delivered to the other party through a secret path and is used once more during decryption. The user will have to properly resolve this error. The third toolbar contains functions used for encryption and decryption.

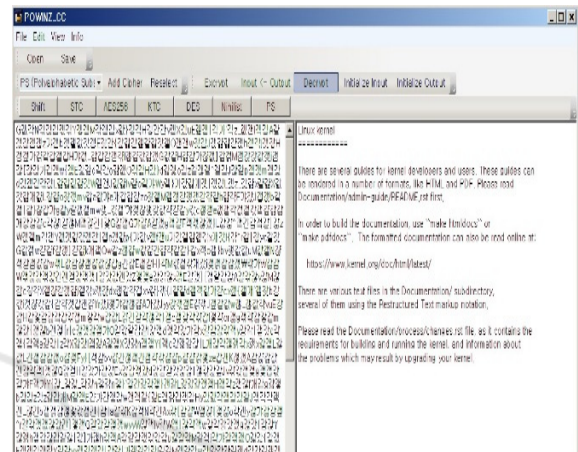


Figure 7: Description of Encrypted Text.

When encryption is complete, this progress bar disappears. The [Input ← Output] button functions to move the encrypted/decrypted text from the output window to the input window. As the text moves to the input window, the text in the output window is erased. Decryption (Description) is performed in the reverse order of the encryption method. In this program, if you enter the same key value as the order of the encryption method entered during encryption, decryption proceeds in the reverse order of encryption. All functions used in this program output logs as shown so that the user can check the functions he or she has used. More detailed logs can be output for the encryption process. However, for information security of the process, only the start and end logs are shown to the user. When the program is started for the first time, a log saying ‘Starting the program’ is output. Each time you use a function of the program, log messages such as ‘Loading a file’, ‘Loading completed’, ‘Shift password added’, ‘STC password added’ are added. Users can also check the encryption method of their choice. Log messages are continuously displayed in every process.

If you press the encryption button after registering all key values, you can encrypt the input text as shown in Figure 6. When all encryption is complete, you can see the encrypted text in the output window. It is also

possible to save the text displayed in the output window and keep it safe in the form of an encrypted file. When all input values are set normally, decryption starts as shown in Figure 7 by clicking the decryption button.

## 5 CONCLUSIONS

The encryption method presented in this paper is not encryption by a single algorithm. Several types of techniques that have already been verified are converged. It is considered that it will be safely used in fields that rely heavily on cryptographic algorithms used in virtual currency and blockchain fields. In the future, encryption-decryption will be more threatened by easily accessible cloud computing than by quantum computers. Later, through the combination of various passwords, functions such as data confidentiality and integrity, authentication and non-repudiation (Nam, Kim, and Park, 2015), as well as stability and robustness of cryptography (Kim, Kim, and Kim, 2002; Song, Ko, and Chung, 2000; Abu-Faraj, and Alqadi, 2020), resistance (Song, Kang, and Sung, 2014; Jang, 2013; Kim, Jang, and Chang, 2014), and efficiency We will continue to study how (Lee, You, and Yim, 2015; Lee, You, and Yim, K. 2016; Kwon, Kim, and Hong, 2014) is improved. In addition, it would be a differentiated significance of this thesis that encryption of documents was easily realized not only in institutions and companies, but also in electronic devices for the general public (Jeon, Shin, Jung, Lee, and Yoo, 2015), which are already becoming common.

## REFERENCES

- Abu-Faraj, M. M., and Alqadi, Z. A. (2020). Using Highly Secure Data Encryption Method for Text File Cryptography, In *IJCSNS International Journal of Computer Science and Network Security*, 20(11):53-60.
- Jang, S. (2013). Design of the File Security Function Using Encryption Algorithm in the Windows Operating System, In *The Korea Institute of Information and Communication Engineering*, 17(3):612-618. (in Korean)
- Jeon, B. Shin, S., Jung, K., Lee, J. and Yoo, K. (2015). Reversible Secret Sharing Scheme Using Symmetric Key Encryption Algorithm in Encrypted Images, In *Journal of Korea Multimedia Society*, 18(11):1332-1341. (in Korean)
- Kim, B., Kim, T. and Kim, J. (2002). FPGA Implementation of the AES Cipher Algorithm by using Pipelining, In *KIISE Transactions on Computing Practices*, 8(6):717-726. (in Korean)
- Kim, T., Jang, M. and Chang, J. (2014). Hilbert-curve based Multi-dimensional Indexing Key Generation Scheme and Query Processing Algorithm for Encrypted Databases, In *Journal of Korea Multimedia Society*, 17(10):1182-1188. (in Korean)
- Kwon, T., Kim, H. and Hong, S. (2014). SEED and ARIA Algorithm Design Methods Using GEZEL, In *Journal of the Korea Institute of Information Security and Cryptology*, 24(1):15-29. (in Korean)
- Lee, K., You, I., and Yim, K. (2015). An Analysis of Agility of the Cryptography API Next Generation in Microsoft: Based on Implementation Example of Applying Cryptography Algorithm HAS-160 in South Korea, In *Journal of the Korea Institute of Information Security and Cryptology*, 25(6):1327-1339. (in Korean)
- Lee, K., You, I., and Yim, K. (2016). An Analysis of a Structure and Implementation of Error-Detection Tool of Cryptography API-Next Generation (CNG) in Microsoft, In *Journal of the Korea Institute of Information Security and Cryptology*, 26(1):153-168. (in Korean)
- Lee, M., Yeom, H., Song, Y., and Kim, J. (2005). *Modern Information Protection*, Korean Publishing Company Hongreung, Seoul. (in Korean)
- Nakov, S. (2018). *Practical Cryptography for Development*, Software University, Sofia.
- Nam, H., Kim, D., and Park, N. (2015). Implementation of ARIA Encryption Algorithm based on WebCL, In *The 42nd Annual Meeting and Proceedings of Korea Computer Conference*, 42(2):49-51, 2015. (in Korean)
- Paar, C., and Pelzl, J. (2010). *Understanding Cryptography: A Textbook for Students and Practitioners*, Springer, Germany.
- Richard E. S., (2001) *Authentication from Passwords to Publish Keys*, Addison-Wesley Professional, Boston. (in Korean)
- Seo, H. (2017). Implementing Software Passwords over the Internet, In *Communications of the Korean Institute of Information Scientists and Engineers*, 35(1):8-15. (in Korean)
- Song, K., Kang, H., and Sung, J. (2014). An Efficient New Format Preserving Encryption Algorithm to Encrypt the Personal Information, In *Journal of the Korea Institute of Information Security and Cryptology*, 24(4):753-763. (in Korean)
- Song, M., Ko, M., and Chung, Y. (2000). Hardware Using of the SEED Algorithm, In *The 24th Conference of the KIPS*, 7(2):1453-1456. (in Korean)