Implementation of RC4 Cryptography Algorithm for Data File Security

A F Doni¹, O A H Maria², S Hanif³
¹,²Informatic Department Engineering, University of Trunojoyo Madura, Indonesia
¹doni.fatah@trunojoyo.ac.id, ²okiemaria@gmail.com, ³hanifudinsukri@trunojoyo.ac.id

Abstract. Communication in the digital age plays an important role, a person can carry out various transactions or exchange data practically and quickly. So as to pose a great risk to information security, ranging from misuse of access or unauthorized authority, modification, change of information, destruction to theft, this is in accordance with the main principles of data and information security ranging from Confidentiality, Integrity, Authentication, and Availability. In research that has been done using RC4 Cryptography Algorithm for Encryption and Decryption in Data Files shows that the RC4 algorithm can run well and is able to secure the authenticity of the data so it is not easily changed by people who are not responsible or people who do not have good access rights in the form text or files in several pdf, doc, Docx, Xls, xlsx or text file formats and directly affect the execution time of encrypted and decrypted files.

Keywords: Encryption, Decryption, Digital, Data Security, Cryptography.

1. Introduction
Communication through the network in the digital era plays an important role, through communication a person can carry out various transactions or exchange data practically and quickly [1]. However, from these developments pose a great risk to information security, ranging from misuse of access or unauthorized authority, modification, change of information, destruction to theft [2][3]. In accordance with the main principles of data and information security starting from (1) Confidentiality of an effort to protect personal data or information from unauthorized persons, (2) Integrity in protecting data or information cannot be changed or modified by unauthorized persons, (3 ) Authentication of efforts to find out the authenticity of data or information from the correct source or service originating from the original source, (4) Availability of business to find out the availability of the system and data or information when needed [4][5][6][7]. data security must be in accordance with the above principles, so as to provide a sense of security for someone in the process of sending data or information obtained according to their intended use. However, a common problem with storing data or information is the issue of confidentiality of file contents [8][9], in a file usually there is information that we consider very important and needs to be kept secure [10].

The easier we get information, the more threats to data security, confidentiality and authenticity. So, to overcome this problem in this study we implement cryptography in the process of securing data or information by disguising it into a encrypted form that has no meaning by using the RC4 cryptographic algorithm for securing data files [11][12][13], with the aim of the data or information that we send can be accepted by real users and in accordance with the original data or information as well as the data or information that is sent is kept confidential from unwanted or unauthorized persons.

2. Methods
a. System Design
In this study generally described using the following block diagram:

![Block diagram of the system](image)

**Figure 1.** Block diagram of the system

Based on Figure 1, the system is built by taking a number of steps starting from uploading the file and selecting the file to do security or encryption, enter the encryption password or encryption key obtained from a 256-bit state-array which is initialized to a key with a length of 1-256 bits \[4][13][14]\. The state-array is randomized to create an encryption key that will be XORed with plaintext or ciphertext then a pseudo-random generation algorithm \[15]\, to generate an encryption key after that the file has been encrypted using a password or key in accordance with what was done before and when the file is run will produce a file that cannot be read, to be able to open the contents of the file must be decrypted by entering the password during encryption previous data if the password or key is wrong then the file cannot be read (ciphertext) but if the password or password is correct then the file will again be read (plaintext)\[12]\.

b. **Cryptography**

Cryptography, in general, is the science and art of maintaining news confidentiality or science that teaches mathematical techniques related to aspects of information security such as data confidentiality, data validity, data integrity, and data authentication \[4][9]\. In cryptography, there are 4 main components that must be fulfilled namely \[11][16][17]\:

1. **The plaintext** is a message or file that can be read.
2. **The ciphertext** is a random message that cannot be read directly.
3. **The key** is the key used to perform cryptographically or security techniques.
4. **The algorithm** method used in the encryption and decryption process.

c. **RC4**

RC4 is one type of stream cipher so that RC4 processes units or input data \[18]\, message or information at one time. This algorithm does not have to wait for a certain amount of input data, messages or information before processing, or add additional bytes to encrypt. The RC4 algorithm has two phases, key setup, and encryption. In the S-bit key setup (S is the length of the key), the encryption key is used to generate an encryption variable that uses two array states (S) and a key (K) \[19]\. An S array of 256 bits is initialized with a number from 0 to 255 \[1][4][13]\. Whereas the 256-bit K array is filled with keys with a length of 1-256 bits repeatedly until the entire K-array is completely filled \[10][20]\. After that, a Key Scheduling Algorithm is performed to produce a permutation of the S array based on the available key and the S-number resulting from the merge operation \[21]\. These merge operations consist of swapping bytes, modulo operations, and other formulas \[13]\.
Figure 2. Flowchart Design Of The RC4 Cryptographic Algorithm Encryption Process
Figure 2 is an RC4 cryptographic algorithm flowchart in the file encryption [4][13], in this study starts from uploading and selecting the file to be introduced then entering the password as a key to support the file in key making is done by changing the characters entered into ASCII code from the code is changed again to decimal numbers after making the XOR Ciphertext process in the file It has been previously selected using the key that was entered and the file that was selected previously was encrypted and the compilation of the file is opened which only looks files that cannot be read or understood, so that the file so that it can be read as it is created is completed the description created in Figure 3.

Figure 3. Flowchart Design of the RC4 Cryptographic Algorithm Decryption Process
In Figure 3 is the design of RC4 cryptographic algorithm flowchart for the file decryption process [4][13], encryption results described in Figure 2, in the decryption process, the encrypted file is uploaded and selected and then entered a password or key used in the encryption process, when the password entered matches the encryption password, a change is made from the characters entered into the ASCII code and from the code is changed again to a decimal number, then the selected and uploaded file is checked, if the file is an encrypted file then the XOR Plaintext process will be carried out in accordance with the key previously entered to be a decrypted file or an unencrypted file but The file entered is not an encryption file, so the decryption process cannot be performed.

3. Result
In this research, the encryption testing file is performed in accordance with the system diagram described in Figure 1. In this process, the type of file to be encrypted is in the form of pdf, doc, Docx, Xls, xlsx or txt files, then enter the password with the number at least 8 characters this password is used to encrypt the process and also during the file decryption process, after the password entry process is complete, the next step is the file encryption process by pressing the Encrypt button on the system, so that the resulting encryption file will be generated when the file is opened will not regular. For more details, see picture 4 below.
From the file encryption process using the RC4 algorithm the encrypted files will be difficult to read and files containing 1 page turn into more than 1 page according to the number of two arrays defined $S$ array totaling 255 initialized from 0 to 255 and $K$ array of 256 bits in length 1-256 bits, for more details, see Figure 5.

After the data file is encrypted, the data cannot be read directly because it has been transformed into irregular data according to the number of characters in the file, the number of pages of the encrypted file also matches the number of characters in the supported file or plaintext. To be able to read the encrypted file or ciphertext the decryption process must be carried out by selecting the encrypted file then entering the password that matches the password used in the encryption process if the password is entered incorrectly then the expected file can be accessed will be read according to the file that is opened or plaintext, for more details can be seen in Figure 6.

The following trial aims to determine the ability of the system to carry out the encryption and decryption process. Testing is done by calculating the time in combination using different key lengths needed from the start of the encryption and decryption process, with the aim to find out the key length significantly affecting performance.
Table 1. Testing Encryption with an 8 character long key

| Testing To | Number of Characters | Data size | Time     |
|------------|----------------------|-----------|----------|
| 1          | 100                  | 12.8 KB   | 0.176 milliseconds |
| 2          | 200                  | 12.9 KB   | 0.12 milliseconds  |
| 3          | 300                  | 13.0 KB   | 0.126 milliseconds |
| 4          | 400                  | 13.9 KB   | 0.21 milliseconds  |
| 5          | 500                  | 14.0 KB   | 0.22 milliseconds  |

Table 2. Testing Encryption with an 16 character long key

| Testing To | Number of Characters | Data size | Time     |
|------------|----------------------|-----------|----------|
| 1          | 100                  | 12.8 KB   | 0.09 milliseconds |
| 2          | 200                  | 12.9 KB   | 0.04 milliseconds |
| 3          | 300                  | 13.0 KB   | 0.18 milliseconds |
| 4          | 400                  | 13.9 KB   | 0.95 milliseconds |
| 5          | 500                  | 14.0 KB   | 0.17 milliseconds |

Table 3. Testing Encryption with an 32 character long key

| Testing To | Number of Characters | Data size | Time     |
|------------|----------------------|-----------|----------|
| 1          | 100                  | 12.8 KB   | 0.08 milliseconds |
| 2          | 200                  | 12.9 KB   | 0.18 milliseconds |
| 3          | 300                  | 13.0 KB   | 0.01 milliseconds |
| 4          | 400                  | 13.9 KB   | 0.14 milliseconds |
| 5          | 500                  | 14.0 KB   | 0.11 milliseconds |

Can be seen in table 1, table 2 and table 3 testing with encryption key lengths of 8, 16, and 32 characters, from the three tables it can be concluded that the key length can affect the speed of encryption and decryption but with a very small time difference, so the difference key length does not affect the overall performance of the encryption and decryption function, and the encrypted file when opened will not display the results but will display an error message “We're sorry. We can't open the name file because we found a problem with is contents”.

4. Conclusion

From the tests that have been done can solve data encryption and decryption using the RC4 algorithm can run well-providing data authenticity so that it is not easily changed by people who are not responsible in the form of text with various formats pdf, doc, Docx, Xls, xlsx or text files. The application of the RC 4 cryptographic algorithm for the encryption and decryption process of data files is supported directly to the file execution time in accordance with the tests that have been carried out. However, it is expected that in future studies, other cryptographic algorithms will be added with other data files such as audio, video, and images.

References

[1] H. M. S. El Hennawy, A. E. A. Omar, and S. M. A. Kholoif, “LEA: Link Encryption Algorithm Proposed Stream Cipher Algorithm,” Ain Shams Eng. J., vol. 6, no. 1, pp. 57–65, 2014.

[2] Q. Qian, Z. ting Yu, R. Zhang, and C. L. Hung, “A multi-layer information dispersal based encryption algorithm and its application for access control,” Sustain. Comput. Informatics Syst., vol. 20, pp. 76–87, 2018.
[3] G. Jaya Krishna, V. Ravi, and S. Nagesh Bhattu, *Key generation for plain text in stream cipher via bi-objective evolutionary computing*, vol. 70. Elsevier B.V., 2018.

[4] R. J. (Bud) Bates, “Cryptography solutions,” *Secur. VOIP*, pp. 83–103, 2014.

[5] Y. Asimi, A. Asimi, A. Guezzaz, Z. Tbatou, and Y. Sadqi, “Unpredictable cryptographic primitives for the Robust Wireless Network Security,” *Procedia Comput. Sci.*, vol. 134, pp. 316–321, 2018.

[6] N. T. Nawal, S A D Houari, “An agent based approach for security integration in Business Rules Management System,” *ICPS ACM Int. Conf. Proceeding Ser.*, no. 6, 2015.

[7] X. Hu and Y. Zhao, “One to One Identification of Cryptosystem Using Fisher’s Discriminant Analysis,” *Int. J. Networked Distrib. Comput.*, vol. 6, no. 3, p. 155, 2018.

[8] N. B. F. Silva, D. F. Pigatto, P. S. Martins, and K. R. L. J. Branco, “Case studies of performance evaluation of cryptographic algorithms for an embedded system and a general purpose computer,” *J. Netw. Comput. Appl.*, vol. 60, pp. 130–143, 2016.

[9] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaied, “Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions,” *Comput. Secur.*, vol. 74, pp. 144–166, 2018.

[10] J. H. Kong, L. M. Ang, and K. P. Seng, “A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments,” *J. Netw. Comput. Appl.*, vol. 49, pp. 15–50, 2015.

[11] S. Sen Gupta, S. Maitra, G. Paul, and S. Sarkar, “(Non-)random sequences from (Non-)random permutations - Analysis of RC4 stream cipher,” *J. Cryptol.*, vol. 27, no. 1, pp. 67–108, 2014.

[12] H. Noura, O. Salman, A. Chehab, and R. Couturier, “Preserving Data Security in Distributed Fog Computing,” *Ad Hoc Networks*, vol. 94, p. 101937, 2019.

[13] P. Jindal and B. Singh, “RC4 encryption - A literature survey,” *Procedia Comput. Sci.*, vol. 46, no. 1cict 2014, pp. 697–705, 2015.

[14] K. G. Paterson and M. Strefler, “A Practical Attack Against the Use of RC4 in the HIVE Hidden Volume Encryption System,” *ASIA CCSACM Symp. Information, Comput. Commun. Secur.*, pp. 475–482, 2015.

[15] I. Muslukhov, Y. Boshmaf, and K. Beznosov, “Source Attribution of Cryptographic API Misuse in Android Applications,” *ASIACCS '18 Proc. 2018 Asia Conf. Comput. Commun. Secur.*, pp. 133–146, 2018.

[16] A. K. Singh, S. G. Samaddar, S. R. Sahoo, and G. Mathew, “Increasing robustness of RC4 family for automated selection of ciphersuites,” *Procedia Eng.*, vol. 30, no. 2011, pp. 45–52, 2012.

[17] K. V. Pradeep and V. Vijayakumar, “Survey on the key management for securing the cloud,” *Procedia Comput. Sci.*, vol. 50, pp. 115–121, 2015.

[18] P. Jindal and B. Singh, “Performance analysis of modified RC4 encryption algorithm,” *Int. Conf. Recent Adv. Innov. Eng. ICRAIE 2014*, vol. 2, no. 28, pp. 0–4, 2014.

[19] D. Bouslimi, G. Coatrieux, M. Cozic, and C. Roux, “Data hiding in encrypted images based on predefined watermark embedding before encryption process,” *Signal Process. Image Commun.*, vol. 47, pp. 263–270, 2016.

[20] E. Taqieddin, O. Abu-Rjei, K. Mhaidat, and R. Bani-Hani, “Efficient FPGA implementation of the RC4 stream cipher using block RAM and pipelining,” *Procedia Comput. Sci.*, vol. 63, no. Euspn, pp. 8–15, 2015.

[21] S. K. Sabnis and R. N. Awale, “Statistical Steganalysis of High Capacity Image Steganography with Cryptography,” *Procedia Comput. Sci.*, vol. 79, pp. 321–327, 2016.