Security Text on Images with RC-128 Bit Symethric Key Encryption

Muslih*[1], **Abdussalam**[2]
Universitas Dian Nuswantoro, Semarang, Indonesia 50131
E-mail : muslih@dsn.dinus.ac.id[1], grey.salam@dsn.dinus.ac.id[2]*
**Corresponding author*

Elkaf Rahmawan Pramudya[3]
Universitas Dian Nuswantoro, Semarang, Indonesia 50131
E-mail : elkaf.rahmawan@dsn.dinus.ac.id[3]

Abstract - The main purpose of using cryptography is to provide the following four basic information security services. One of the purposes of cryptography is secrecy. Confidentiality isthe fundamental security service provided by cryptography. This is a security service that stores information from unauthorized persons. Confidentiality can be achieved through a variety of ways ranging from physical security to the use of mathematical algorithms for data encryption. Vernam cipher is a stream cipher where the original data or plain with 8x8 block operation. Experimental results prove that RC4 can perform encryption and decryption with a fast execution process. In this study used a processor with 8GB of RAM. The encryption result of the text used yields the average encryption time and decryption average of 2 seconds.

Keywords – MRI, Tumor identification, Image segmentation, K-Means, Fuzzy C-Means

1. INTRODUCTION

Information security is a real concern and protects data from various risks, especially data theft. Protection of data has been done a lot, but with the development of types and media of data dissemination, it is necessary to have special techniques that collaborate with technology in order to save data. Some examples of data security have been set on ATM machines, use of passwords on credit cards, passwords on email addresses, social media and data protection at certain agencies or institutions. An institution certainly has a data storage warehouse which is often referred to as a server. In this case, server security can be determined from the data protection operations performed. Data security, especially in a server is very important.

Data protection techniques that have existed since the ancient era and are still used and proven safe are cryptography [1]. Cryptography itself according to its development, it can be seen that the use of keys must be adjusted to the input data used. There are various types of cryptography [2] namely symmetrical, asymmetrical, hash functions [3]; where these types of cryptography can be grouped again based on their appearance, namely classical and modern [4]–[6].

The use of keys in symptomatic cryptography is the private key only, so it is more secure. Since its inception, Caesar Cipher as one of the symptomatic cryptographic algorithms has been used [2], [7], [8]. Until now, various kinds of symptomatic key algorithms have developed between ROT13, Vernam Cipher [9]–[12], Blowfish [13], AES [14]–[16], One Time Pad (OTP) [12], [17], [18], RC4 [19], and so on.

In this paper, it will be explained further about the results of experiments that have been carried out in analyzing the RC4 operating model which is the development of RC1, RC2 and RC3.

2. RESEARCH METHOD

1.1. State of the Art

Related to the excellence of cryptography, especially in the RC algorithm, the following is the state of the art that underlies this research as brief in Table 1.

Table 1. State of the Art

Year	Researchers	Algorithm	Explanation
2006	Allam Mousa; Ahmad Hamad [20]	RC4	The RC4 parameter has indicated that the encryption speed or decryption time is directly related to the encryption key and the data file size if the data is large enough.
2014	Nishith Sinha, et. al [21]	RC4	RC4 Development with Product Cipher. With optimization, RC4 becomes faster and data length becomes longer.
2016	Isnar Sumartono, et. al [22]	RC4	This study states that RC4 has good performance, is fast and can be used on long plaintext
2016	Rajni Tiwari; Amit Sinhal [23]	RC4	It only uses text data, but RC4 is proven to be a fast algorithm with a high time execute value.

1.2. Cryptography

Cryptography based on the type of key used can be divided into symmetric and asymmetric cryptography. A symmetric key means that the keys used in the encryption and decryption process must be exactly the same, whereas in the asymmetric key type, it uses two keys known as the private key and the public key. The use of symptomatic and asymmetric keys is often related to the types of ciphers used, for example stream ciphers and block ciphers. For example, stream ciphers are like the RC4 algorithm that we have implemented, while the block cipher is for example RC6.

1.3. RC4

RC4 was developed by Ron Rivest who is also known as the Rivest Cipher 4. Here the stream cipher is used for plain text encryption [20]. The Pseudorandom bit stream (key stream) is generated by the RC4 algorithm, and bitwise encryption / decryption has been performed. The generation key system involves two stages, namely [19], [21], [23]:

1. Permutation 256 bytes.
2. Sub-block with an 8-bit index pointer.

The key length for this RC4 is between 40-128 bits. If common block ciphers are not used with strong MAC, bitflapping attacks are possible and stream-cipher attacks are also vulnerable if not implemented properly.

```
j = 0
 for i = 0 to 255
 j = j + S[i] + K[i]swap (S[i],
 S[j])
```

Encryption and decryption is done by XORing the bytes from the plaintext / ciphertext with random bytes from the S-box to produce ciphertext / plaintext, as follows :

Initialize i and j to zero

For each byte of plaintext (or ciphertext):

```
i = i + w
    j = k + S [j + S[i]]k = i + k + S[j]
    swap (S[i], S[j])
    z = (S[j + S[i + S[z+k]]]
Decryption: plaintext [i] = S[z] XOR ciphertext [i]Encryption: ciphertext [i] =
S[z] XOR plaintext [i]
```

Byte K, that has been XORed with plaintext to produce ciphertext or XORed to produce plaintext. RC4 encryption is extremely fast, approximately 10 times faster than DES. The steps for the RC4 encryption algorithm used in this scientific paper are as follows:

1. Get the data to be encrypted and the selected button.
2. Create two string arrays.
3. Starting an array with numbers from 0 to 255.
4. Fill in another array with the selected button.
5. Randomize the first array depending on the array of buttons.
6. Scrambles the first array within itself to generate the final main stream.
7. XOR the final stream with the data to be encrypted to provide the cipher text.

2.3. Proposed Method

The following are the stages of research that we have carried out in conducting experiments on Cryptography with the RC4 algorithm.

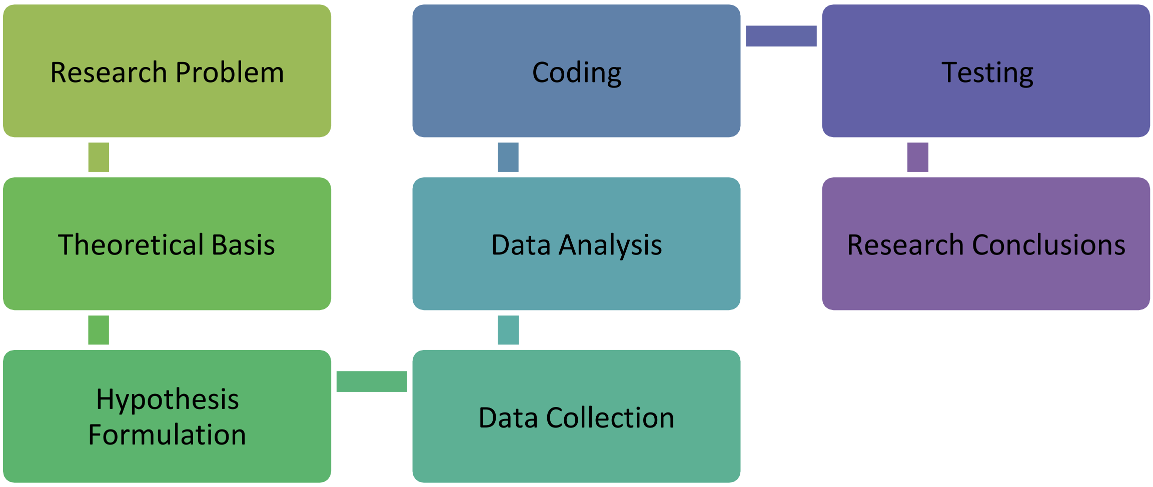


Figure 1. Proposed Method

3. RESULTS AND DISCUSSION

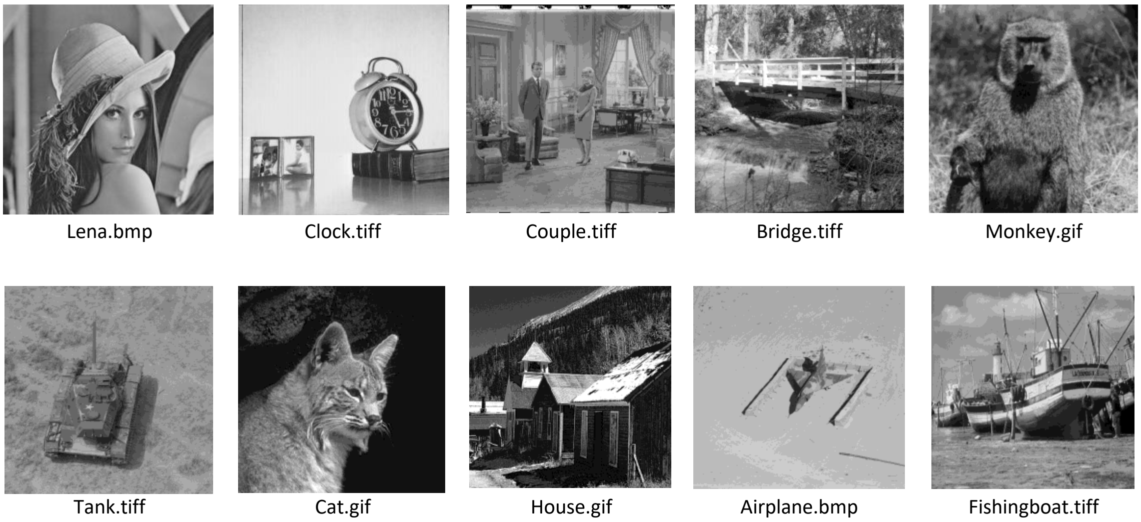
RC4 which is generally used for text data protection, in this study has been implemented on image media. In this study, grayscale images are used from http://sipi.usc.edu/database/ where this url is also used by various other studies that use image media such as watermarking and steganography implementation. The number of images that we tested was 10 images. Each image has been encrypted using a different key in order to know the time taken from the RC4 operation via Matlab. The following is a pseudocode that we created for the encryption and decryption process:

```
key='kriptografimemangtepatuntukmelakukanproteksidata';for k = 1:n
        i = 1 + mod(i + 1, 256);
        j = 1 + mod(j + S(i), 256);tmp = S(i);
        S(i) = S(j);
        S(j) = tmp;
        Ks = 1 + mod(S(i) + S(j), 256);
        EI(k) = bitxor(S(Ks), I(k));
end
```

To evaluate the results of encryption and decryption, this research uses the calculation of entropy and time execute. A digital image can be recognized based on the special characteristics contained in the image, one of which is by using the entropy value (e). Entropy can be used to measure the randomness or uncertainty of an object, which can be applied in cryptography. If the entropy value of the decrypted image with the original image is the same, the image quality is good, which indicates that the image resulting from the cryptographic decryption process has returned to its original form as in (1).

$$Entropi = -\sum_{i=1}^{n} p_i \, Log_2 \, p_i \qquad (1)$$

Based on (1), *pi* is the color value of the pixel. Another measurement will be done by comparing the original image histogram with the encrypted and decrypted histogram.



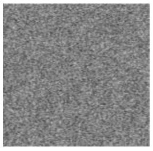
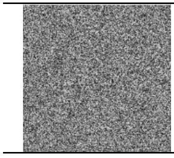
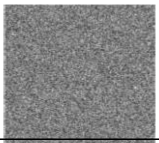
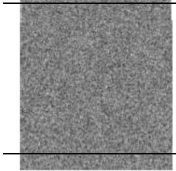
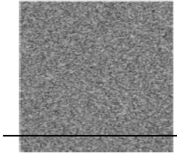
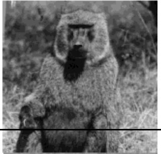
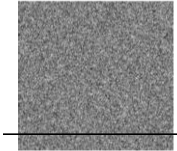
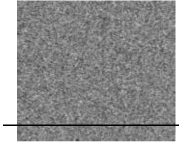
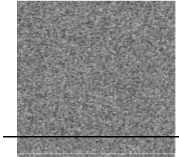
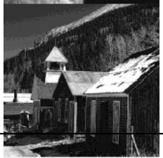
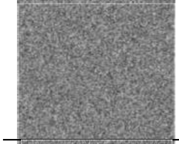
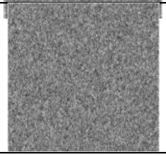
Lena.bmp Clock.tiff Couple.tiff Bridge.tiff Monkey.gif

Tank.tiff Cat.gif House.gif Airplane.bmp Fishingboat.tiff

Figure 2. Dataset in Grayscale images in size 512x512 pixels

Testing evaluation using Figure 2, shown in Table 2.

Table 2. Results of entropy and time taken

Image	Encryption	Decryption	Entropy	Time taken (s)
Lena.bmp			7.9993	0.049406
Clock.tiff			7.9969	0.016004
Couple.tiff			7.9993	0.029815
Bridge.tiff			7.9993	0.035212
Monkey.gif			7.9994	0.037880
Tank.tiff			7.9994	0.037880
Cat.gif			7.9994	0.029365
House.gif			7.9994	0.029275
Airplane.bmp			7.9992	0.030070
Fishingboat.tiff			7.9993	0.035563

Based on Table 2 and Figure 2 above, it can be seen that all images have been successfully encrypted and decrypted correctly. Proof of encryption is that the image is in the form of "salt and peper" or in plain view the image has been damaged. However, the decryption process proves that the encrypted image can be extracted back into the initial image as before it was processed through encryption. For the entropy value, the entire image produces values above 7. The range of entropy values is between 0 and 1. The lowest and highest entropy values are not too far apart. Meanwhile, the time taken for execution (time taken) is between 0.02 and 0.03 seconds. The following is a summary of the entropy values from Table 1 and 10 other images that we have executed.

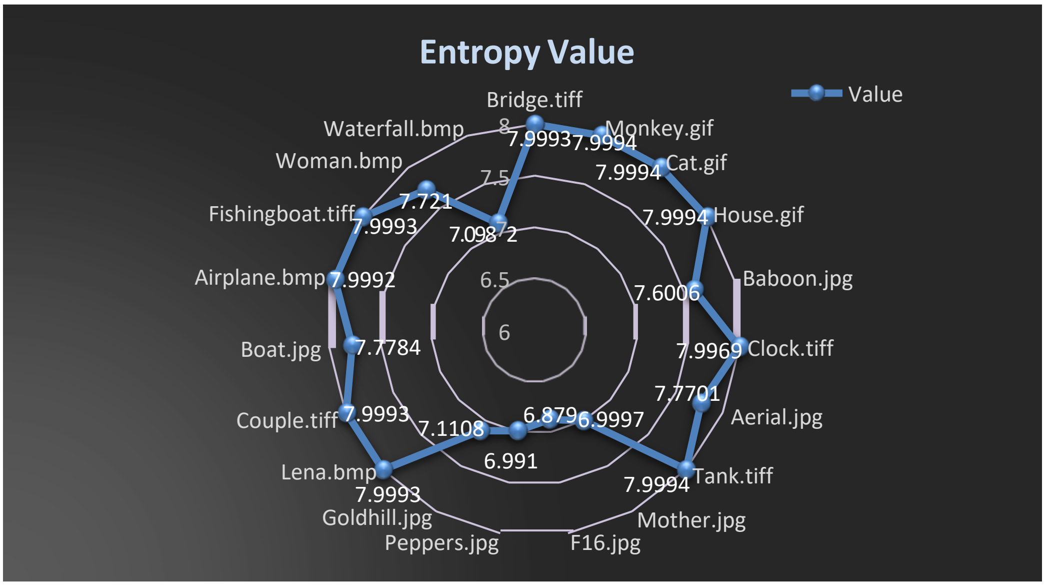
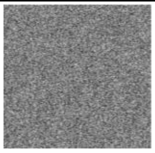
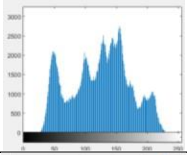
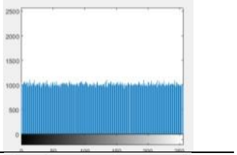
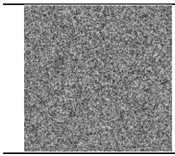
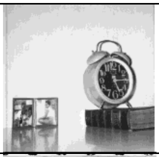
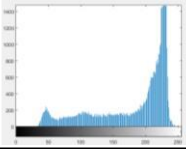
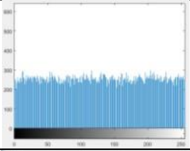
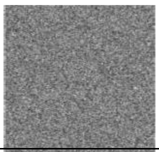
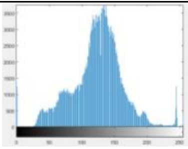
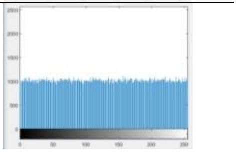
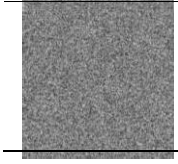
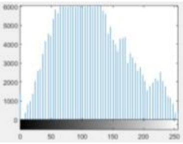
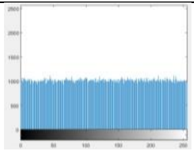
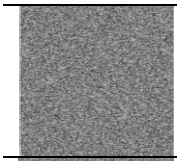
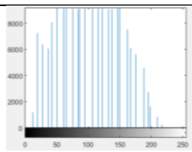
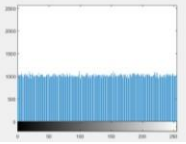
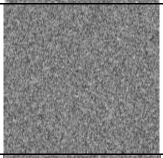
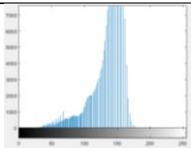
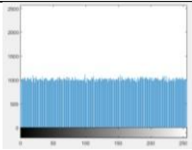
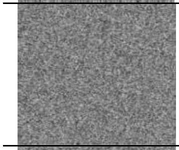
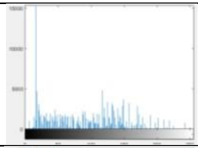
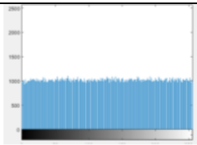
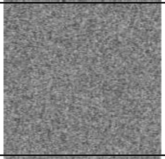
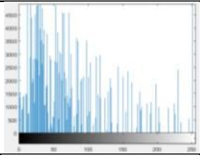
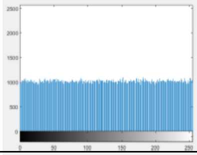
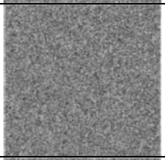
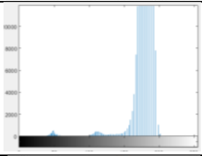
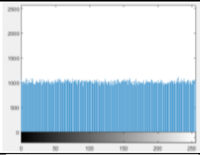
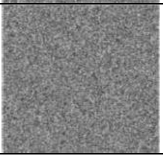
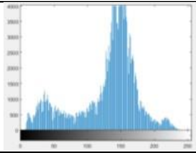
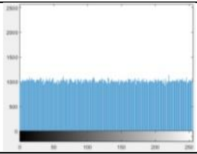


Figure 3. Entropy Values

Table 3. Encryption and Decryption Histogram

Image	Encryption	Decryption	Encryption Hostogram	Decryption Histogram
Lena.bmp				
Clock.tiff				
Couple.tiff				
Bridge.tiff				

Monkey.gif				
Tank.tiff				
Cat.gif				
House.gif				
Airplane.bmp				
Fishingboat.tiff				

Based on Table 3 and Figure 4, it can be concluded that the encrypted histogram is different from the original histogram. On the other hand, the decrypted image is the same as the original image histogram. Thus the decryption image can be read back correctly without any change in pixels. To find out more about the difference between unencrypted and encrypted images, it can be seen from the surface and mesh of the image.

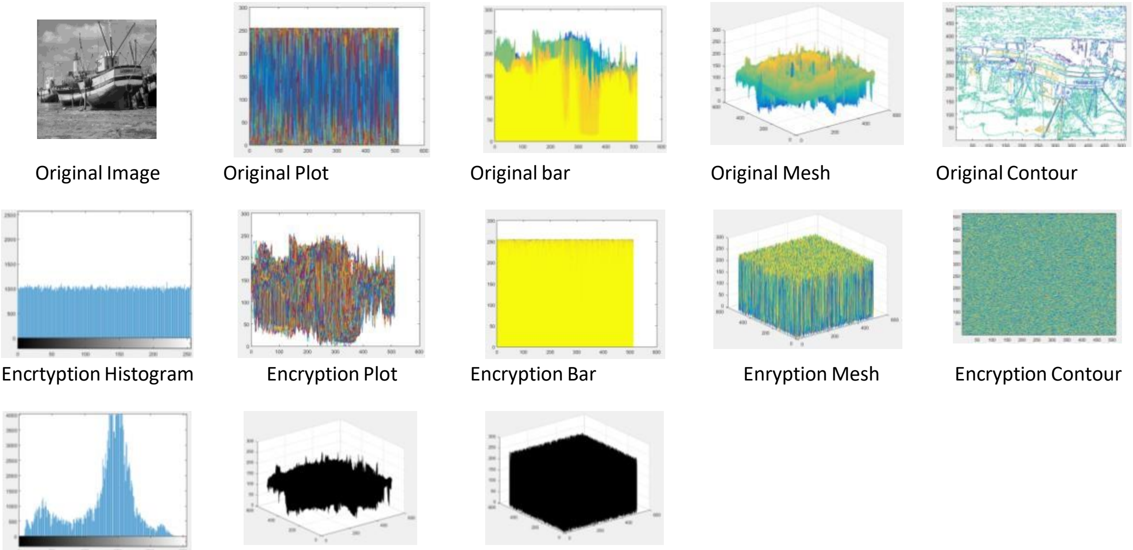

Original Image
Original Plot
Original bar
Original Mesh
Original Contour
Encrtyption Histogram
Encryption Plot
Encryption Bar
Enryption Mesh
Encryption Contour
Decryption Histogram
Original Surf
Enryption Surf

Figure 4. A Comparison between several display based on encryption and decryption

4. CONCLUSION

The key should not be used more than one, otherwise the attacker could reduce the encrypted message and get the combination of the two unencrypted messages. Encryption is a way to secure files that are owned by someone. Data cannot be opened other than by the owner. One way to secure it is using cryptography. In this study, a type of symmetric cryptographic algorithm is used, namely RC4-128 bit, which is the development algorithm of RC2 and RC3. The private key used is in the form of text with a length of 1776 bits. For operations and experiments, we used Matlab 2015, while the test results used entropy, time taken, and histogram. Based on the tests we have done, it can be concluded that the file format affects the entropy value and the length of time tested. This can be seen in the .jpg file format used, the average entropy gain is lower than the .bmp, .tiff, and .gif file formats. for the travel time required for the encryption process is counted fast between 0.01 to 0.03 seconds. In this test we used an Intel ® Core ™ i7-7500U CPU with 8192 MB of memory.

REFERENCES

[1] K. R. Ilaga and C. A. Sari, “Analysis of Secure Image Crypto-Stegano Based on Electronic Code Book and Least Significant Bit,” *J. Appl. Intell. Syst.*, vol. 3, no. 1, pp. 28–38, 2018.

[2] E. H. Rachmawanto and C. A. Sari, “Keamanan File Menggunakan Teknik Kriptografi Shift Cipher,” *Techno.COM*, vol. 14, no. 4, pp. 329–335, 2015.

[3] K. Ganeshkumar and D. Arivazhagan, “Generating a digital signature based on new cryptographic scheme for user authentication and security,” *Indian J. Sci. Technol.*, vol. 7, no. Specialissue6, pp. 1–5, 2014.

[4] V. Kumar and P. K. Koul, “Robust RSA for Digital Signature,” *IJCSI Int. J. Comput. Sci. Issues*, vol. 8, no. 6, pp. 359–362, 2011.

[5] P. Saha, “A comprehensive study on digital signature for internet security,” *Accent. Trans. Inf. Secur.*, vol. 1, no. 1, pp. 1–6, 2016.

[6] C. J. Mitchell, “On the Security of 2-Key Triple DES,” *IEEE Trans. Inf. Theory*, vol. 62, no. 11, pp. 6260–6267, Nov. 2016.

[7] M. B. Pramanik, “Implementation of Cryptography Technique using Columnar Transposition,” pp. 19–23, 2014.

[8] K. Senthil, K. Prasanthi, and R. Rajaram, “A modern avatar of Julius Ceasar and Vigenere cipher,” in *2013 IEEE International Conference on Computational Intelligence and Computing Research*, 2013, pp. 1–3.

[9] E. Rachmawanto, C. Sari, Y. Astuti, and L. Umaroh, “KRIPTOGRAFI VERNAM CIPHER UNTUK MENCEGAH PENCURIAN DATA PADA SEMUA EKSTENSI FILE,” in *PROSIDING SEMINAR NASIONAL MULTI DISIPLIN ILMU & CALL FOR PAPERS UNISBANK (SENDI_U) KE-2 Tahun 2016*, 2016, pp. 46–51.

[10] M. Jain and S. K. Lenka, “Secret data transmission using vital image steganography over transposition cipher,” in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, 2015, pp. 1026–1029.

[11] A. Saraswat, C. Khatri, Sudhakar, P. Thakral, and P. Biswas, “An Extended Hybridization of Vigenere and Caesar Cipher Techniques for Secure Communication,” *Procedia Comput. Sci.*, vol. 92, pp. 355–360, 2016.

[12] O. Tornea, M. E. Borda, V. Pileczki, and R. Malutan, “DNA Vernam Cipher,” *Proc. 3rd Int. Conf. E-Health Bioeng. - EHB 2011*, pp. 24–27, 2011.

[13] R. Mathur, S. Agarwal, and V. Sharma, “Solving security issues in mobile computing using cryptography techniques &amp;#x2014; A Survey,” in *International Conference on Computing, Communication & Automation*, 2015, pp. 492–497.

[14] P. C. Mandal, “Superiority of Blowfish Algorithm,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 2, no. 9, pp. 196–201, 2012.

[15] D. P. Joseph and M. Krishna, “Cognitive Analytics and Comparison of Symmetric and Asymmetric Cryptography Algorithms,” *Int. J. Adv. Res. Comput. Sci.*, vol. 6, no. 3, pp. 51–56, 2015.

[16] Sangeeta and E. A. Kaur, “A Review on Symmetric Key Cryptography Algorithms,” *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 4, pp. 358–362, 2017.

[17] V. Rekhate, “Secure and Efficient Message Passing in Distributed Systems using One-Time Pad,” in *2016 International Conference on Computing, Analytics and Security Trends (CAST) College of Engineering Pune, India. Dec 19-21, 2016 Secure*, 2016, pp. 393–397.

[18] R. Shukla, H. O. Prakash, R. P. Bhushan, S. Venkataraman, and G. Varadan, “Sampurna Suraksha: Unconditionally Secure and Authenticated One Time Pad Cryptosystem,” in *2013 International Conference on Machine Intelligence and Research Advancement*, 2013, pp. 174–178.

[19] A. D. Irfianti, “METODE PENGAMANAN ENSKRIPSI RC4 STREAM CIPHER UNTUK APLIKASI,” *Semin. Nas. Apl. Teknol. Inf. 2007 (SNATI 2007)*, vol. 2007, no. Snati, p. 4, 2007.

[20] A. Mousa and A. Hamad, “Evaluation of the RC4 Algorithm for Data Encryption,” *Int. J. Comput. Sci. Appl.*, vol. 3, no. 1, pp. 44–56, 2006.

[21] N. Sinha, M. Chawda, and K. Bhamidipati, “Enhancing Security of Improved RC4 Stream Cipher by Converting into Product Cipher,” *Int. J. Comput. Appl.*, vol. 94, no. 18, pp. 17–21, May 2014.

[22] I. Sumartono, A. P. U. Siahaan, and N. Mayasari, “An Overview of the RC4 Algorithm,” *IOSR J. Comput. Eng.*, vol. 18, no. 6, pp. 67–73, 2016.

[23] R. Tiwari and A. Sinhal, “Block based text data partition with RC4 encryption for text data security,” *Int. J. Adv. Comput. Res.*, vol. 6, no. 24, pp. 107–113, May 2016.