Scientific Journal of Informatics
Vol. 5, No. 2, Nov 2018

p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Cryptography Triple Data Encryption Standard (3DES) for Digital Image Security

Christy Atika Sari[1], Eko Hari Rachmawanto[2], Christanto Antonius Haryanto[3]

[1,2,3]Informatic Engineering Departement, Computer Science Faculty, Universitas Dian Nuswantoro
Email: [1]atika.sari@dsn.dinus.ac.id, [2]eko.hari@dsn.dinus.ac.id, [3]christantoantonius@gmail.com

Abstract

Advances in technology in the field of internet, making the Internet become the most popular data transmission media for now. This is certainly not free from the incidence of cyber crime, such as theft and data modification. Given the losses caused by data manipulation is very detrimental to the owner, then the original data can be misused in the cyber world. Cryptography offers an algorithm for randomizing data, so it can not be read by unauthorized people. The cryptography technique was implemented using Triple Data Encyption Standard (3DES) given the results of a randomized cryptographic algorithm, it is possible to arouse suspicion from the viewer. For that will be done the process of insertion of cryptographic files into another media in the form of images commonly referred to as steganography. The steganography technique that will be used is End of File (EOF). The combination of 3DES and EOF in the 64x64 pixel image with grayscale color format produces the fastest image processing time of 173.00192 seconds with the highest Peak Signal to Noise Ratio (PSNR) value of 25.0004 dB, while in the 128x128 pixel image with grayscale format has produced the highest PSNR 21.0084 dB.

Keywords: Cryptography, Triple Data Encryption Standard, End of File, Peak Signal to Noise Ratio

1. INTRODUCTION

Improved technology in the field of internet trigger the occurrence of problems, especially in the field of Internet security. Given the number of Internet media users are quite high, it is possible the occurrence of Internet crimes that occur in the process of sending data through the internet media. Among these are file modifications, or the theft of important data transmitted over the internet. For solving cryptography offers a way to scramble an existing file by using a specific key known only to the user. The technique of cryptography that will be implemented in this research is 3DES. 3DES has a good level of security when compared to other cryptographic techniques [1]. However, cryptography has a weakness, the resulting file looks random, and of course this can cause suspicion for people who see it [2]. For overcoming these deficiencies need to be done the process of insertion of the encrypted file into another media, which is often referred to as steganography.

The technique of steganography used is EOF. The EOF technique is chosen because it has a fast time [3] compared to other steganographic techniques.

The combination of both methods will be implemented on a 64x64 pixel digital image with a grayscale color format. As for the insertion media that will be used on the method EOF in the form of image 'pepper.png' with grayscale color format measuring 512x512 pixels. Based on the above problem, the purpose of this research is the implementation of 3DES cryptography algorithm and EOF steganography to secure image data sent via internet media.

2. METHODS

2.1. Cryptography

Cryptography has actually existed since 4000 years ago in Egypt, which we often know with hieroglyph. In Ancient Rome it was told that Julius Caesar wanted to send a message to someone on the battlefield, but he did not want the message to be known to others [4]. Therefore, Julius Caesar created a way to hide the message, from the problem then created the algorithm Caesar Cipher. In cryptography there are 2 types of encryption models are:

a) Symmetrical: Where in the process of encryption or decryption will use the same key.
b) Asymmetric: Encryption and decryption process using different keys.

The cryptographic algorithm consists of three basic functions, namely [5]:

a) Encryption: Is a process of securing data, by converting the original text into code that is not understood based on the existing algorithm.
b) Key: Is the key that will be used to perform the process of encryption and decryption.
c) Decryption: It is the process of recovering the previously encrypted code back into the original text, using algorithms and keys that have been used previously.

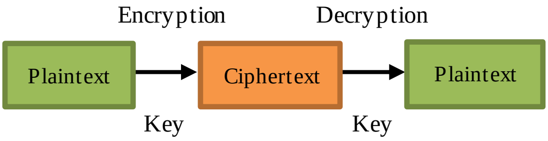


Figure 1. Common cryptography scheme [6]

Based on Figure 1, the encryption process will be performed using plaintext and keys that are processed by a particular algorithm. With this step will be generated a cipher file, which is ready to be sent. For the decryption process, it will be done with the cipher text and the initial key is processed with the initial algorithm.

2.2. Steganography

Steganography is a science or art in hiding information by entering the information into other media [3]. So that the existence of the message cannot be known by others. Steganography comes from the Greek Steganós [7] which means to hide and Graptos which means writing, so the whole meaning is "writings that are hid". Media commonly used in steganography include:

Image: .bmp, .gif, .jpeg, .jpg, .tiff, .png.
Audio: .wav, .mp3
Text: .rtf, .txt, .pdf, .doc, .docx.

The purpose of steganography is to conceal the existence of a data, by hiding it into a particular media.

2.3. Triple Data Encryption Standard (3DES)

3DES is a symmetric cryptography algorithm with cipher block type. Symmetric cryptography is a method that will use the same key in the encryption and decryption process. Cipher block is a type of asymmetric cryptography that has a fixed or fixed bit size, which is 64 bits for DES [8]. 3DES is the development of Double DES algorithm previously available to improve DES security. The 3DES algorithm uses three keys in the encryption and decryption process. The key variations in 3DES can be classified into 3 by using the same 1 key, 2 different keys, or 3 different keys to each other. 3DES encryption using 2 or 3 different keys is still considered robust for current usage [9].

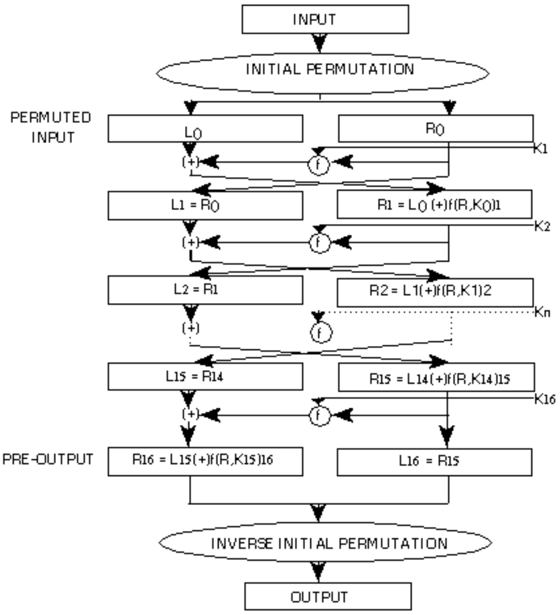


Figure 2. 3DES scheme using permutations

Based on Figure 2 above it can be seen that the sequence of 3DES algorithm scheme. Encryption formulas for 3DES can be described as follows:

$$\mathrm{C} = \mathrm{E3}_{\mathrm{k1,k2,k3}}(\mathrm{P}) = \mathrm{E}_{\mathrm{k3}}(\mathrm{D}_{\mathrm{k2}}(\mathrm{E}_{\mathrm{k1}}(\mathrm{P}))) \quad (1)$$

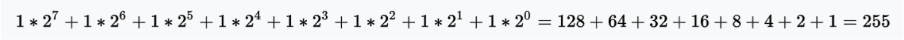
Where $E3_{k1\,,k2,k3}(P)$ is 3DES encryption for P using key, are k_1, k_2, and k_3. Whereas $E_{kn}\,(P)$ is DES encryption for P using k_n. For $D_{kn}(P)$ is DES decryption for P using key k_n. The decryption formula for 3DES can be described as follows:

$$P = \ D3_{k1\,,k2,k3}(C) = D_{k1}\,(E_{k2}\,(D_{k3}\,(C))) \quad (2)$$

Where $D3_{k1\,,k2,k3}\,(P)$ 3DES decryption of C using key, are k_1, k_2, and k_3. Whereas $D_{kn}\,(P)$ is DES decryption for P using k_n. For $E_{kn}(P)$ is encryption for P using key $k_{n.}$

2.4. Least Significant Bit (LSB)

The LSB is part of the binary number row that has the least or meaningless value. The location of LSB is the rightmost bit sequence [10]. For example, in a byte 0101 1001 the LSB bit is the bit located at the far right of "1". Sometimes the letter "b" at the end of the number becomes 0101 1001b.



$$1 * 2^7 + 1 * 2^6 + 1 * 2^5 + 1 * 2^4 + 1 * 2^3 + 1 * 2^2 + 1 * 2^1 + 1 * 2^0 = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$$

Figure 3. Example of LSB payload calculation

From the 1st row above, the 1st rightmost digit is 1, and that is the smallest. The section is called the least significant bit (the least significant bit), while the leftmost part is 128 and is called the most significant bit (Joshi & Yadav 2015). Least significant bits are often used for the purposes of insertion of data into another digital medium (Gosalia 2016). Suppose the message to be inserted 5 bits = 11010, then the number of bytes used = 5 bytes 10010110 11001001 11111001 10001000 10100011 (byte used for message insertion) The insertion process message 11010, the insertion result becomes 10010111 11001001 11111000 10001001 10100010, so this LSB method is only replace the first bit.

2.5. Peak Signal to Noise Ratio (PSNR)

PSNR is the result of comparison of the maximum value of the image measured by the noise. PSNR in this study will be used to perform the process of comparison of results from images before and after the steganography process. The higher the PSNR value is similar to the image between the comparable images [11]. The formula of the PSNR value can be defined according to equation (3) and equation (4) below.

$$PSNR = \ 10log_{10}\left(\frac{C^2 max}{MSE}\right) \quad (3)$$

With the value of MSE (Mean Square Error) obtained according to the formula as follows:

$$MSE = \frac{1}{M.N}\sum_{X=1}^{M}\sum_{Y=1}^{N}(S_{xy} - C_{xy})^2 \tag{4}$$

2.6. Histogram

An image histogram is a graph depicting the distribution of pixel intensity values of an image[12], or just a part of an image. In the cipher file histogram if it looks flat for each color intensity is the same and this indicates that the algorithm used cannot provide any guidance for a statistical attack [13] because there is no prominent intensity as in plain file histogram.

3. RESULTS AND DISCUSSION

Here, we proposed a scheme both encryption and decryption process as present in Figure 4.

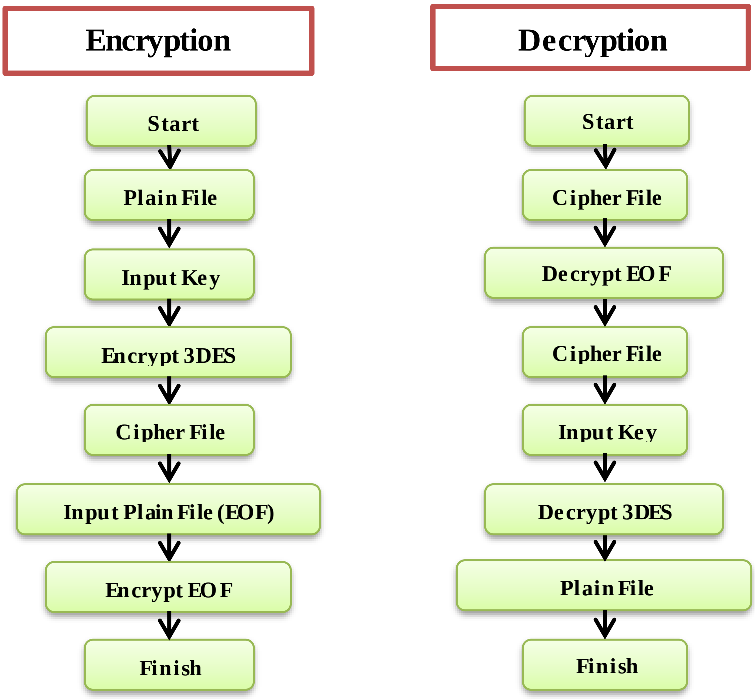


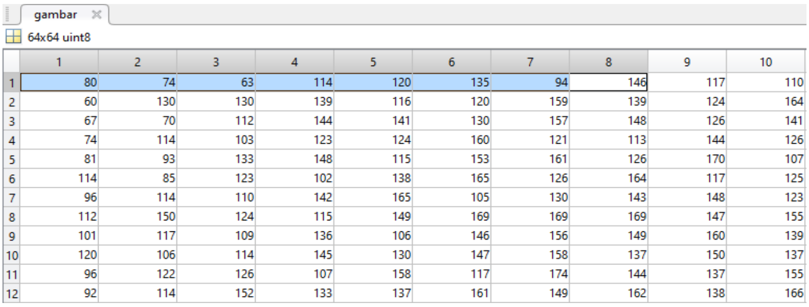
Figure 4. Proposed scheme 3DES-EOF

Based on Figure 4 can be explained the description as follows. For the encryption step, image-shaped data to be tested is first prepared in advance. Input the plain file to be encrypted, then will be asked input in the form of the key as much as 3 pieces of plain text 8 characters that will be inputted to the program to encrypt the image with 3DES. Then the cipher file generated from the encryption will be used on the EOF encryption. Previously will be asked for input image that will be used for the process of encryption EOF. After encryption EOF is done will get image file result of cipher file 3DES and EOF.

For the process of decryption itself, the first requested input in the form of cipher files 3DES and EOF. Then will be done by using encryption algorithm EOF. After that will be requested input in the form of 3 pieces of key previously used to perform the encryption process of 3DES. After that the 3DES algorithm will run, and generate plain files that can be reused. The image to be used as a sample of this encryption process is the baboon.png that can be seen in Figure 5 with the size of 64x64 pixels which has the grayscale color format.

Figure 5. Baboon.png

For the value to be used for the calculation is the first 8 pixels contained in the image baboon.png. The pixels of baboon shown in Figure 6.



	1	2	3	4	5	6	7	8	9	10
1	80	74	63	114	120	135	94	146	117	110
2	60	130	130	139	116	120	159	139	124	164
3	67	70	112	144	141	130	157	148	126	141
4	74	114	103	123	124	160	121	113	144	126
5	81	93	133	148	115	153	161	126	170	107
6	114	85	123	102	138	165	126	164	117	125
7	96	114	110	142	165	105	130	143	148	123
8	112	150	124	115	149	169	169	169	147	155
9	101	117	109	136	106	146	156	149	160	139
10	120	106	114	145	130	147	158	137	150	137
11	96	122	126	107	158	117	174	144	137	155
12	92	114	152	133	137	161	149	162	138	166

Figure 6. Pixels of baboon.png

From Figure 6 above it can be seen that the pixel value of the first 8 pixels in baboon.png is:

80 74 63 114 120 135 94 146

Once the pixel value is obtained, the pixel will be converted to binary, the binary value to be generated is:

80	74	63	114
01010000	01001010	00111111	01110010

120	135	94	146
01111000	10000111	01011110	10010010

After obtaining the binary value of 8 pixels early baboon.png, will be processed into the binary value of the key that will be used in 3DES algorithm. The key to be used consists of 8 characters. The key to be used for the 3DES algorithm in this study is 'COMPUTER'. Of the 8 characters will be processed into binary values using ASCII table. The following is the result of the binary key value 'COMPUTER':

C	O	M	P
01000011	01001111	01001101	01010000
U	T	E	R
01010101	01010100	01000101	01010010

After obtained both the input image and key values, will be the iteration of 3DES calculation. For encryption in plain file, the first step to do is processing by using the initial permutation table in Figure 7, after initial permutation is generated, the value of the existing bit will be split into 2 parts ie L0 and R0.

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Figure 7. Initial permutation table of DES

As for the encryption on the key, will be done by using the PC-1 to produce CD (k) table. The value of each resulting binary will be split into C0 and D0 which shown in Figure 8.

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	45	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Figure 8. Table PC-1 of DES

Then the results of C0 and D0 will be processed using the Left Shift table which can be seen in Figure 9.

Iteration	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Left Shift	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Figure 9. Left shift iterations

The result of the shift by using Left Shift table will be called C1 and D1, then the result will be processed using PC-2 table as shown below in Figure 10.

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Figure 10. Table PC-2 of DES

At this stage, the key is finished and ready for use in the final calculation process. For the next step, the processing will be done on the plain files that have been processed earlier by using the expansion table. Then the result of the expansion table will be XOR with the result from the PC-2 table as follow in Figure 11.

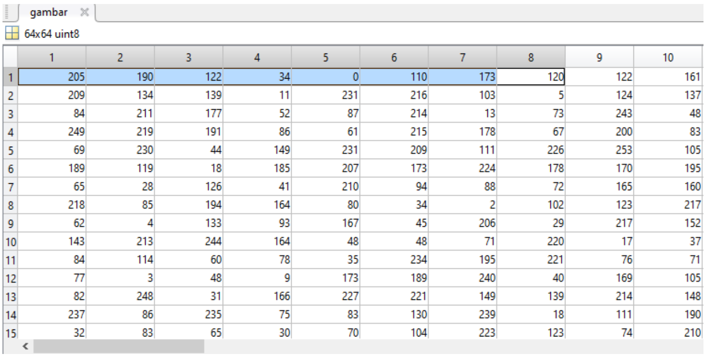
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Figure 11. Expansion table of DES

After the XOR process will be obtained binary value of 48 bits to be processed using table S-Box and P-Box. This step is a one-time iteration of the 3DES algorithm using the first key. After processing using 3 pieces with 16 different iterations called L16 and R16, the following binary values are obtained bellow and Figure 12.

205	190	122	34
L_{16} : 11001101	10111110	01111010	00100010

0	110	173	120
R_{16} : 00000000	01101110	10101101	01111000



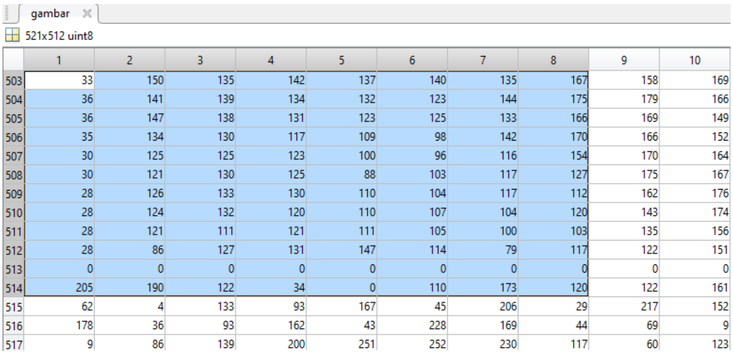
	1	2	3	4	5	6	7	8	9	10
1	205	190	122	34	0	110	173	120	122	161
2	209	134	139	11	231	216	103	5	124	137
3	84	211	177	52	87	214	13	73	243	48
4	249	219	191	86	61	215	178	67	200	83
5	69	230	44	149	231	209	111	226	253	105
6	189	119	18	185	207	173	224	178	170	195
7	65	28	126	41	210	94	88	72	165	160
8	218	85	194	164	80	34	2	102	123	217
9	62	4	133	93	167	45	206	29	217	152
10	143	213	244	164	48	48	71	220	17	37
11	84	114	60	78	35	234	195	221	76	71
12	77	3	48	9	173	189	240	40	169	105
13	82	248	31	166	227	221	149	139	214	148
14	237	86	235	75	83	130	239	18	111	190
15	32	83	65	30	70	104	223	123	74	210

Figure 12. A pixels of encryption result

The process of the EOF algorithm has been done by inserting the pixel value at the end in an image used as a medium. Results from the EOF process. The image to be used in the EOF process in this study is the image of 'peppers.png' measuring 512x512 pixels in grayscale color format as shown in Figure 13.

Figure 13. Peppers.png used in EOF process after DES

Based on the Figure 14, can be seen that the value of the results of 3DES calculation has been successfully inserted in the image by using the algorithm EOF. Image data to be used in this study as many as 10 pieces of images taken from petitcolas.net [9] with description, 5 images processed size 64x64 pixels and 128x128 pixels. The image to be used has a .png and .bmp format. For the encryption process will be done using 3DES cryptography algorithm and EOF steganography.



	1	2	3	4	5	6	7	8	9	10
503	33	150	135	142	137	140	135	167	158	169
504	36	141	139	134	132	123	144	175	179	166
505	36	147	138	131	123	125	133	166	169	149
506	35	134	130	117	109	98	142	170	166	152
507	30	125	125	123	100	96	116	154	170	164
508	30	121	130	125	88	103	117	127	175	167
509	28	126	133	130	110	104	117	112	162	176
510	28	124	132	120	110	107	104	120	143	174
511	28	121	111	121	111	105	100	103	135	156
512	28	86	127	131	147	114	79	117	122	151
513	0	0	0	0	0	0	0	0	0	0
514	205	190	122	34	0	110	173	120	122	161
515	62	4	133	93	167	45	206	29	217	152
516	178	36	93	162	43	228	169	44	69	9
517	9	86	139	200	251	252	230	117	60	123

Figure 14. A result of DES-EOF

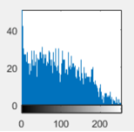
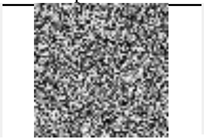
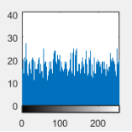
The time comparison between images with the size of 64x64 pixels with a 128x128 pixel image shown in Table 1. It appears that the fastest processing time of a 64x64 pixel image is 172.00139 seconds, while for 128x128 images it is 680.99123 seconds.

Table 1. Time elapse of 3DES-EOF

No	Name of Image	Size of Image	Time Elapse (in second)	
			Encryption	Decryption
1	baboon.png	64x64	179.27272	174.29913
2	bear.bmp	64x64	176.28819	175.29316
3	f16.png	64x64	174.19215	179.29913
4	lochness.bmp	64x64	177.99314	178.99182
5	papermachine.png	64x64	173.00192	172.00139
6	baboon.png	128x128	680.99123	683.20013
7	bear.bmp	128x128	685.00134	689.99132
8	f16.png	128x128	688.11324	681.33045
9	lochness.bmp	128x128	683.04834	686.91344
10	papermachine.png	128x128	689.13775	683.22134

The result of 3DES-EOF ont the encryption process shown in Table 2 and Table 3. PNSR value of EOF is shown in Table 4.

Table 2. Result of 3DES-EOF on the encryption process 64x64 pixels

Name of Image	Plain File	Histogram	Cipher File	Histogram
baboon.png				

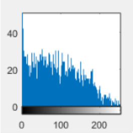
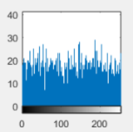
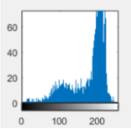
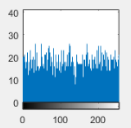
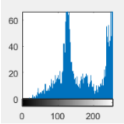
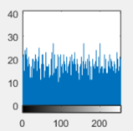
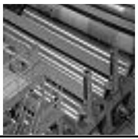
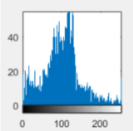
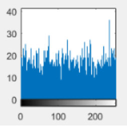
bear.bmp				
f16.png				
lochness.bmp				
papermachine.png				

Table 3. Result of 3DES-EOF on the encryption process 128x128 pixels

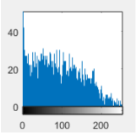
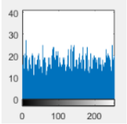
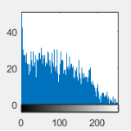
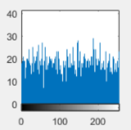
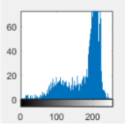
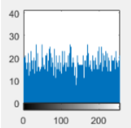
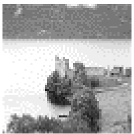
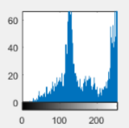
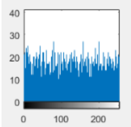
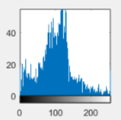
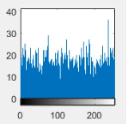
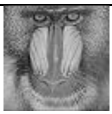
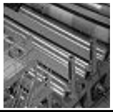
Name of Image	Image	Histogram	Cipher File	Histogram
baboon.png				
bear.bmp				
f16.png				
lochness.bmp				
papermachine.png				

Table 4. PSNR value of EOF

Name of Image	Plain File	EOF 64x64 pixels		EOF 128x128 pixels	
		Stego Image	PSNR (db)	Stego Image	PSNR (db)
baboon.png			24.9788		21.0081
bear.bmp			25.0702		20.9198
f16.png			24.9302		20.9793
lochness.bmp			25.0549		21.0301 dB
papermachine.png			25.0004		21.0084

Based on Table 4, it can be seen that the resulting PSNR values vary from 20 dB to 25 dB for the tested image. We tested several plain file using peppers.png as shown in Figure 13.

4. CONCLUSION

The combination of both algorithms has been successfully applied to grayscale images of 64x64 pixels and 128x128 pixels with image format .png and .bmp. With the lowest travel time for 64x64 pixel images is 172.00139 seconds and 680.99123 seconds for images with 128x128 pixels in size. For the resulting PSNR values vary between 20 dB to 25 dB. Suggestions that can be given for further research is the application can be implemented with the use of GUI with other image format as well as in color image format.

5. REFERENCES

[1] Thakur, J., & Kumar, N. (2011). DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis. *International Journal ofEmerging Technology and Advanced Engineering*, *1*(2), 6-12.

[2] Cheddad, A., Condell, J., Curran, K., & Mc Kevitt, P. (2010). Digital Image Steganography: Survey and Analysis of Current Methods. *Signal processing*, *90*(3), 727-752.

[3] Cole, E. (2003) *Hiding in Plain Sight: Steganography and the Art of. Indianapolis, Indiana*, Canada: Wiley Publishing.

[4] Ardy, R. D., Indriani, O. R., Sari, C. A., & Rachmawanto, E. H. (2017, November). Digital Image Signature Using Triple Protection Cryptosystem (RSA, Vigenere, and MD5). In *Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS), 2017 International Conference* on (pp. 87-92). IEEE.

[5] Sari, W. S., Rachmawanto, E. H., & Sari, C. A. (2017). A Good Performance OTP Encryption Image Based on DCT-DWT Steganography. *Telkomnika*, *15*(4), 1987-1995.

[6] Kusuma, E. J., Indriani, O. R., Sari, C. A., & Rachmawanto, E. H. (2017, November). An Imperceptible LSB Image Hiding on Edge Region Using DES Encryption. In *Innovative and Creative Information Technology (ICITech), 2017 International Conference on* (pp. 1-6). IEEE.

[7] Ardiansyah, G., Sari, C. A., & Rachmawanto, E. H. (2017, November). Hybrid Method using 3-DES, DWT and LSB for Secure Image Steganography Algorithm. In *Information Technology, Information Systems and Electrical Engineering (ICITISEE), 2017 2nd International conferences on* (pp. 249-254). IEEE.

[8] Nilesh, D., & Nagle, M. (2014, January). The New Cryptography Algorithm with High Throughput. In *Computer Communication and Informatics (ICCCI), 2014 International Conference on* (pp. 1-5). IEEE.

[9] Kaur, A. (2017). A Review on Symmetric Key Cryptography Algorithms. International Journal of Advanced Research in Computer Science, *8*(4): 358-362.

[10] Watters, P., Martin, F., & Stripf, H. S. (2008). Visual Detection of LSB-Encoded Natural Image Steganography. *World Wide Web Internet And Web Information Systems*, *5*(1): 24–32.

[11] Almohammad, A., & Ghinea, G. (2010, July). Stego Image Quality and the Reliability of PSNR. In *Image Processing Theory Tools and Applications (IPTA), 2010 2nd International Conference on* (pp. 215-220). IEEE.

[12] Rachmawanto, E. H., & Sari, C. A. (2017, October). A Performance Analysis Stegocrypt Algorithm Based on LSB-AES 128 bit in Various Image Size. In *Application for Technology of Information and Communication (iSemantic), 2017 International Seminar on* (pp. 16-21). IEEE.

[13] Li, X., Zhang, W., Ou, B., & Yang, B. (2014, July). A Brief Review on Reversible Data Hiding: Current Techniques and Future Prospects. In *Signal and Information Processing (ChinaSIP), 2014 IEEE China Summit & International Conference on* (pp. 426-430). IEEE.