Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 8, August 2017, pg.133 – 142

Expanded 128-bit Data Encryption Standard

Bryan F. Cruz[1], Keinaz N. Domingo[2], Froilan E. De Guzman[3], Jhinia B. Cotiangco[4], Christopher B. Hilario[5]

[1]Department of Computer Studies and Systems, University of the East, Caloocan Philippines
[2]Department of Computer Studies and Systems, University of the East, Caloocan Philippines
[3]Department of Computer Studies and Systems, University of the East, Caloocan Philippines
[4]Department of Computer Studies and Systems, University of the East, Caloocan Philippines
[5]Department of Computer Studies and Systems, University of the East, Caloocan Philippines

[1] bryancruz014@gmail.com; [2] keinazd6@gmail.com; [3] froilanic@yahoo.com; [4] jhiniaaa04@yahoo.com; [5] chris.hilario0108@gmail.com

Abstract— This paper presents a 128-bit approach on the outdated Data Encryption Standard cipher. Since the symmetric block cipher is well past its prime, many methods have been devised by hackers in order to crack the cipher and obtain the plaintext message, namely through brute force attacks. In order to improve its security, the authors have made modifications to the standard bit size, wherein it is doubled from a size of 64-bits to 128-bits on the key structure and plaintext block. The size of various tables, functions, keys and swaps that are found throughout the process of the original DES are also subject to this doubling in size. Henceforth, the Expanded Data Encryption Standard is twice as extensive as its predecessor. By increasing the overall size of the cipher, it will take much longer for an attacker to bypass the security through the use of brute force.

Keywords— Cryptography, Symmetric Block Cipher, Data Encryption Standard, Key Scheduling, Substitution Box

I. INTRODUCTION

In the modern age where technology has become a major part of most of the people due to the conveniences it brings, it also brings with it a possible breach of privacy or confidentiality, such as messages between two parties being intercepted and abused by an outsider. This necessitate a need of an adequate and effective cryptographic algorithms to secure these kind of data transmissions from an unauthorized user revealing. [4] Although the act of data encryption can be used by private individuals, national security is still the predominant motive for data, encryption. [20] To successfully design & implement security we need to be a step ahead or perhaps think on the same line as the cyber criminals do. [17] Preserving secrecy against outsiders is an issue that has been prevalent in the past, and this is where cryptography plays a part in helping to keep exchanges private messages.

Cryptography covers a wide area of techniques, including those such as microdots, merging words with images, and other ingenious ways to hide information from being intercepted by outside parties. It is the process of converting messages from a comprehensive form into an incomprehensive one at one end and which reverses the process at the other end so that the message is unreadable by interceptors or eavesdropper without the secret knowledge. [16]

Though in modern times, cryptography is most known or associated with scrambling plaintext, which is the plain message that is going to be sent in transit into cipher text. Of the various ciphers, block ciphers are the type often used in data encryption. This is because academic research in block ciphers has progressed along a

different course than research in stream ciphers.[19] Block cipher papers have traditionally been concrete designs (with specific parameters and names) or breaks of those designs. [13] Block cipher is an encryption algorithm that has inputs and outputs as much as 1 block. Each block is generally comprised of 64 bits or 128 bits. [2] Many blocks ciphers have been conceived, such as the RSA cipher, among many others. Although there is one cipher that is part of the first few to be invented.

An example of a symmetric block cipher is the Data Encryption Standard (DES), which is one of the oldest symmetric key-algorithms. This algorithm was published for public comment in March 1975, after undergoing Government review for acceptability as a Federal standard. [5] The DES algorithm for encryption and decryption, which is the main theme of this lecture, is based on what is known as the Feistel structure. [3] It was officially standardized in 1976 becoming the first encryption system to meet the National Bureau of Standards (NBS) criteria for an encryption system [8] and the first standardized encryption system. The publication of DES was fundamental in the public understanding of modern block cipher design. It operates by using 64-bit blocks along with a 56-bit key. Going through the process of DES, wherein there are 16 rounds to be processed, first you need to encrypt the plaintext or the message, second you need a key generator to change the key for every round.

Unfortunately, in 1999 DES was declared as not safe enough, mostly because of the small key size that was inadequate when the technology advancement is considered. [11] To remedy that, in this paper, the authors have devised a way to improve the general security of this block cipher. By increasing the original 64-bit blocks to an amount of 128-bit blocks, along with raising the 56-bit key to a 112-bit key, it will improve the protection of the cipher against brute force attacks by increasing the amount of time needed to crack the cipher.

II. LITERATURE REVIEW

Cryptography is the application of various scientific techniques in order to encode messages and give them a layer of security against outside parties who are not meant to view the message. By applying cryptography into important exchanges of information, it prevents unscrupulous individuals from abusing the information they could receive. One study related to cryptography states the following: Different encryption techniques are used to protect the confidential data from unauthorized use, [6] which is the essence of cryptography, and some of these cryptographic techniques fall into the category of being symmetric block ciphers.

In the field of cryptography, a symmetric block cipher is used in order to encrypt a given plaintext into a cipher text. Through the use of such methods, it allows different parties to exchange messages between each other without the risk of a leak of confidentiality. Symmetric block ciphers rely on the use of plaintexts, blocks of bits, and keys. The strength of encryption algorithm heavily relies on the computer system used for the generation of keys, [18] which are used to encrypt and decrypt the message. The method of using blocks and keys is common between symmetric block ciphers, such as DES or the Data Encryption Standard.

DES is one of the first block ciphers to be invented, although it no longer sees much use due to its outdated nature. A survey (Wiener, 2001) shows the time it takes for cryptanalyst to break cryptographic algorithms. In 1999, a distributed net project broke a DES key in 23 h using exhaustive key search method. [8] Due to there being several weaknesses, it is not used as often as it used to be. As a result, many researchers have made studies in order to modify and improve the original.

Different approaches have been made in modifying the DES algorithm. One research states the aim of the authors: This research aims to fuse DES algorithm with Blowfish algorithm and Genetic Algorithm (GA). [9] The Blowfish algorithm is another symmetric block cipher, while the Genetic algorithm is a method for solving constrained and unconstrained optimization problems. Adding additional forms of algorithms to the base algorithm would then introduce more layers of defence. Next is the GMDES, wherein the authors state the following: The proposed algorithm is a modified version of DES which uses graph Hamiltonian cycle and the graph automorphism concept for generating keys. [14] The strength of the algorithm is that it didn't fully depend on the secret key.

A study named NEWDES was stated to be simpler than the original by the authors. It does not use initial and final permutations. All operations are on entire bytes – at no time does the algorithm read any particular bits. The central f-function is much, much simpler than in DES. They key in NEWDES is longer than in DES: 120 bits or 15 bytes. [10] The overall structure of the new cipher was similar to that of DES, but it was straightforward to program in a high-level language, and it yields programs that run quickly. [10]

Another study uses a new manipulation bits process has been added in by using different truth table for manipulation bits process work on 4-states (0,1,2,3), while the traditional binary process (XOR) work on (0, 1) bits only. [15] The authors of the cited study stated that the security would be increased, along with adding additional key, modified S-Box design, modifies function implementation and replacing the old XOR. [15] One study chose to make changes to the permutations, by making them dynamic in nature. Dynamic permutation is intended to permute the transposition and substitution matrices (IP, IP^{-1}, S-Boxes, PC1, PC2, Expansion matrix). [1] By doing so, one of the weaknesses of the DES would be solved.

The authors of a paper based on the hybrid use of DES and RSA applied their research to Bluetooth technology. The speeds of DES encryption is up to several M per second, it is suitable for encrypting large

number of message. Seeing from key management, RSA algorithm is more superior than the DES algorithm. Because the RSA algorithm can distribute encryption key openly, it is also very easy to update the encryption keys. [12] By combining the strengths of RSA and DES, the authors of the paper found a way to better secure Bluetooth.

One paper in particular shows a similarity to ours, in that the study also used 128 bit keys and blocks. It was made in order to protect against brute force attacks. The first problem of bruit force attack is here resolved by making variable size key that is variable in size it depends on the size of key provided in the form of any text, picture, audio or video. [21] Since DES is weak against brute force attacks, introducing variable sizes into the cipher could confuse an attacker.

III. DATA ENCRYPTION STANDARD ALGORITHM

Data Encryption Standard (DES) is a block cipher that relies on encryption techniques of confusion and diffusion [1] that uses plaintext blocks with the size of 64-bits. It also returns the blocks of cipher text in the same size. These 64-bit blocks are then divided into two blocks with 32-bits each. These blocks are the left half L and the right half R. The original DES algorithm is shown on Figure 1.

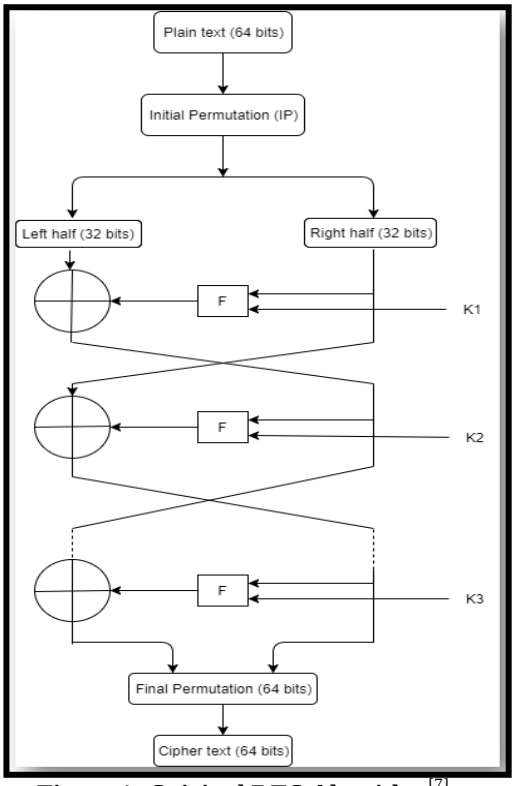

Figure 1: Original DES Algorithm[7]

M = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
L = 0000 0001 0010 0011 0100 0101 0110 0111
R = 1000 1001 1010 1011 1100 1101 1110 1111

Figure 2: 64-bit plain text [22]

Along with the plaintext as shown on Figure 2, DES uses key sizes of 56-bits on the 64-bit blocks. Every 8th bit is ignored, which results in a 56-bit key size. The bits are still numbered from 1 to 64, going from the left to the right. The 64-bit key is permuted using the Permuted Choice-1 (PC-1). After going through the PC-1, only 56-bit of the original key will be used in the new permuted key. This key is now split into two, which is left and right halves, then each half has 28 bits. They are then put through a circular left shift, which is dependent on the assigned number of left shifts for that round. The value obtained from the shift is then used as the input for PC-2 in order to produce a 48-bit output. For the message, it goes through a table which is called the Initial Permutation (IP). The message data that consists of 64-bits are rearranged according to the IP. For example, the 58th bit of M becomes the first bit of IP. This process is repeated for the rest of the numbers in the IP. Once the output is obtained, the permuted block is divided into two halves, each consisting of 32 bits.

Along through the next process, it goes through 16 loops, for 1<=n<=16, using a function f which performs on two blocks, a data block of 32 bits and key K_n of 48 bits to produce a block of 32 bits. This will result to a final block, for n = 16, of $L_{16}R_{16}$. That is, in each loop, the right 32-bit of the previous result will be taken and swapped with left 32-bit of the current step. For the right 32-bit in the current step, the left 32-bit of the previous step are XORed with the calculation f. To calculate for f, expand each block R_{n-1} from 32-bit to 48-bit. This is done by using a selection table then repeats some of the bits in R_{n-1}. The use of this selection table for the purpose of this explanation will be called function **E**. Thus $\mathbf{E}(R_{n-1})$ has a 32-bit input block, and 48-bit output block. Such that E be the 48 bits of its output, written as 8 blocks of 6 bits each, are obtained by selecting the bits in its inputs in accordance to a table named the E-bit selection table.

The calculations for the function f are still not done. After the previous steps, an output of 48-bits or eight groups of six bits is obtained. These groups of bits are used as an input for the S-box, as a means to further process the bits. These groups of six bits will give an address in a different S-box. In that address will be a 4-bit number, and this number will replace the original 6-bits. The result after going through the S-box is that the eight groups of 6-bits are transformed into the same number of groups, but are now 4-bits, for a total of 32-bits.

The final stage in the computation of f is to do work on permutation **P** of the **S**-box output to obtain the final value of f:

$$f = \mathbf{P}(S_1(B_1)\ S_2(B_2)\ ...S_8(B_8))$$

P yields a 32-bit output from a 32-bit input by permuting the bits of the input block.

In the next round, an output of $L_2 = R_1$ is the product, which came from the block that was previously calculated, upon receiving that result, calculate for $R_2 = L_1 + f\ (R_1, K_2)$, and so on for 16 rounds. After the sixteenth round, the blocks L_{16} and R_{16} are now obtained. The arrangement of the two blocks is then reversed into the 64-bit block, $R_{16}L_{16}$ and a final permutation named $\mathbf{IP}^{-1}$ will be applied. Upon applying the final permutation, the encrypted form of the original message is the end result.

Decryption is the reciprocal of encryption, by following the same steps as above, but reversing the order in which the same sub keys will be applied.

IV. PROPOSED WORK

In Figure 3 shows the block diagram of the 128-bit DES. It's similar to the block diagram that the original uses, except it now uses 128-bit plaintext and a 128-bit key. The 32-bit swap after round 16 is now a 64-bit swap.

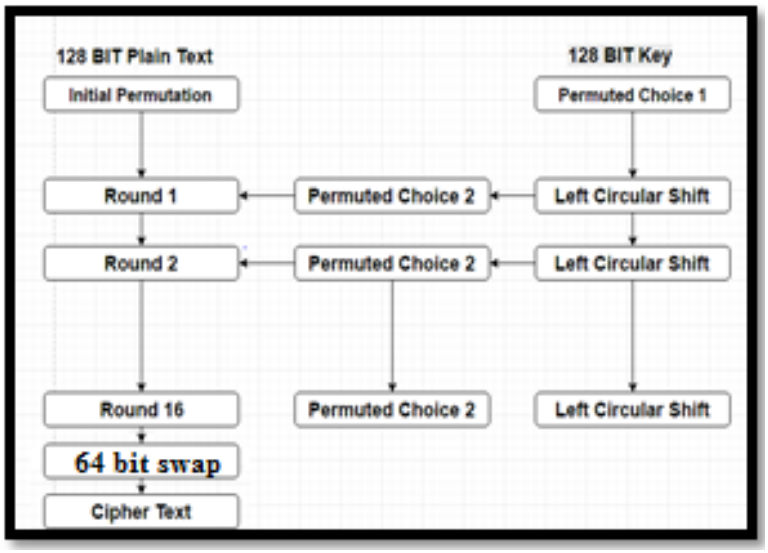

Figure 3: Block Diagram of Expanded 128-Bit DES

The process of the Expanded 128-bit DES remains mostly the same, except for some special cases such as the various tables that need to be expanded in order to account for the increase in the amount of bits from 64-bits to 128-bits. Table 1 shows the 128-bit Permuted Choice 1. Whereas the original DES used a table with a size of 7x8, the new PC-1 uses a 7x16 table. Due to the increase in the size of the tables, dividers have been placed to for the isolation of the two parts of the permutation.

Table 1: Modified Permuted Choice 1 (PC-1)

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	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4
121	113	105	97	89	81	73
65	122	114	106	98	90	82
74	66	123	115	107	99	91
83	75	67	124	116	108	100
127	119	111	103	95	87	79
71	126	118	110	102	94	86
78	70	125	117	109	101	93
85	77	69	92	84	76	68

After the 128-bits have been permuted with the modified PC-1, it goes through the original steps of the 64-bit DES, which is to shift the bits depending on the current round. Once the bits have been shifted to the left, it is used as an input for the Table 2, which is the expanded and modified PC-2. Compared to the original PC-2 that uses 48-bits, the Modified PC-2 uses 96-bits.

Table 2: Modified Permuted Choice (PC-2)

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
70	73	67	80	57	61	59	84
71	62	77	66	79	75	68	60
82	64	72	63	83	76	69	58
97	108	87	93	103	111	86	96
107	101	89	104	100	105	95	112
90	109	102	98	106	92	85	88

Round Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits Rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Figure 4: Modified Key Shifts

Similarly, to the 64-bit DES, the 128-bits also go through an initial permutation with a total number of 128-bits.

Table 3: Modified Initial Permutation

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
122	114	106	98	90	82	74	66
124	116	108	100	92	84	76	68
126	118	110	102	94	86	78	70
128	120	112	104	96	88	80	72
121	113	105	97	89	81	73	65
123	115	107	99	91	83	75	67
125	117	109	101	93	85	77	69
127	119	111	103	95	87	79	71

Figure 5 shows the modified I-th round function of the 128-bit DES. It shows some similarities to the standard DES, but some parts have been changed to account for the new set of bits to be used. The first 32-bits have been doubled to 64-bits. This doubling in the amount of bits also applies to the rest of the bits that are going to be used later on. For example, the original 48-bits have become 96-bits, while the number of the 32-bits is now twice that, becoming 64-bits.

Table 4: Substitution Box (Sbox)

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S1/S9	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S2/S10	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	3	13	18	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S3/S11	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	15	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S4/S12	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S5/S13	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S6/S14	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S7/S15	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S8/S16	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Shown in the table above is the Substitution box to be used with the Expanded DES. The overall structure of the S-box is the same as that of the S-box found in the standard DES, except it is now used twice because of the expansion of the bit size. S1 is used as S9, S2 as S10, so on and so forth.

Table 5: Modified Expansion Table

64	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	33
32	33	34	35	36	37
36	37	38	39	40	41
40	41	42	43	44	45
44	45	46	47	48	49
48	49	50	51	52	53
52	53	54	55	56	57
56	57	58	59	60	61
60	61	62	63	64	1

Above is the modified Expansion table for the Expanded DES. Compared to the original Expansion table which consisted of 48 bits, it has been expanded and now contains a total of 96 bits.

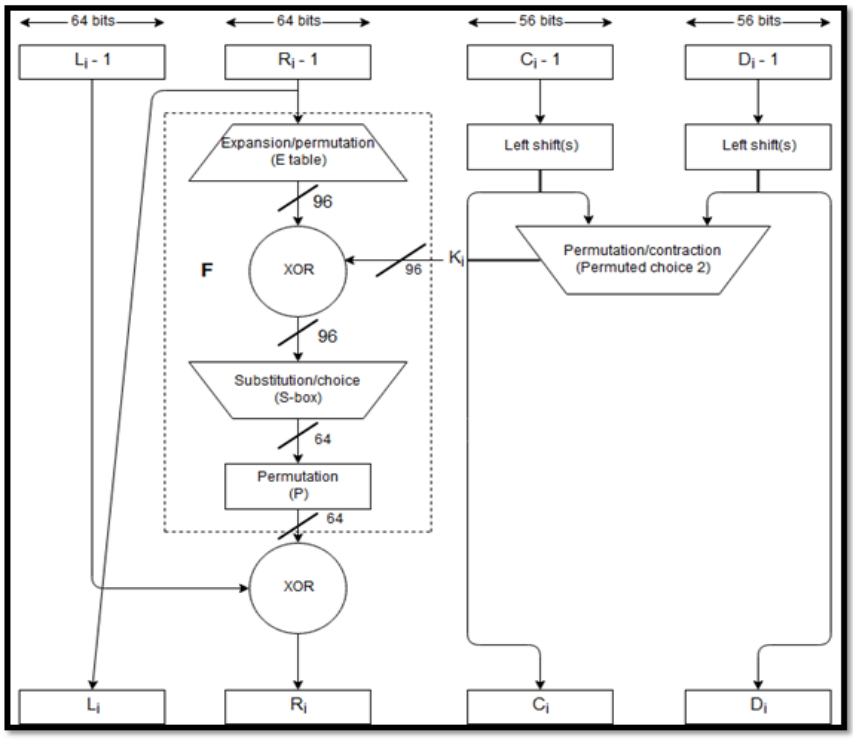

Figure 5: I-th Model of 128-Bit DES

V. RESULTS AND DISCUSSION

In order to make use of the new 128-bit DES, it is necessary to make use of the new tables that have been constructed. A step-by-step demonstration, along with examples has been documented below by the authors.

First, the message is to be inputted, and converted into binary. After that, the different keys that are going to be used for every round are produced. Next, the bits are permuted with the Modified Permuted Choice 1, after which the circular left shift is performed on the bits. Once the bits have been shifted, they are permuted again, but this time with the use of the Modified Permuted Choice 2 so that the message can be encrypted. After 16 keys have been, the Modified Initial Permutation is used on the message and it is split into two parts. The right half goes through the Expanded Permutation to expand it. Once that is done, it is necessary to bitXOR it with the Permuted Choice 2 results that were obtained from the key. Next, the Sbox is used to substitute for the

assigned values according to the addresses. Afterwards, it goes through another round of another permutation in order to reduce the value. Lastly, the value that was obtained is bitXORed with the left half of the message in order to obtain the first round or R1 of the plaintext. To get the value of L1, simply get the value of R-1.

To successfully encrypt the message, a key is needed for every round. For the 128-bit DES, a 128-bit key is used. Table 6 is an example of how the keys are generated for every round, from round 1 to round 16.

Table 6: Key Generation of 128 DES

K1 = E00A6E5724C52BC352DEC4F83972
K2 = E00A6E5724C52A86A5BD89F072E5
K3 = 8029B95C9314AB1A96F627C1CB96
K4 = 00A6E5724C52AE6A5BD89F072E58
K5 = 029B95C9314AB8A96F627C1CB961
K6 = 0A6E5724C52AE0A5BD89F072E586
K7 = 29B95C9314AB8096F627C1CB961A
K8 = A6E5724C52AE005BD89F072E586A
K9 = 4DCAE498A55C01B7B13E0E5CB0D4
K10 = 372B92629570O5DEC4F83972C352
K11 = DCAE498A55C0147B13E0E5CB0D4B
K12 = 72B926295700S3EC4F83972C352D
K13 = CAE498A55C014DB13E0E5CB0D4B7
K14 = 2B9262957005370F83972C352DE
K15 = AE498A55C014DC13E0E5CB0D4B7B
K16 = 5C9314AB8029B927C1CB961A96F6

After going through all of the steps, it results in the first round of the 128-bit DES, which is shown in the figure 6 below.

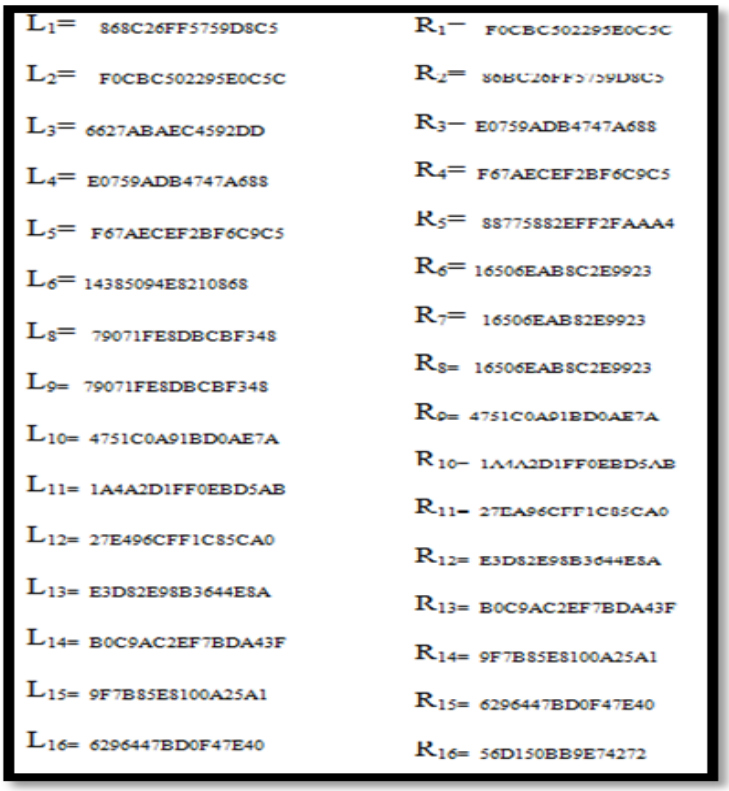

Figure 6: Output of 128 DES

Once the message is encrypted and it is needed to return it into the original plaintext, it can be decrypted. The first step is to get the latest ciphertext, then get the permutation for that specific round and bitXOR it with the right. The output will be the left cipher-text of round 15. To get the right round 15, get the value from the

left and put it on the right half. After going through 16 rounds of decryption, process it through the inverse initial permutation to get the decrypted ciphertext. The figure below is the process of the decryption.

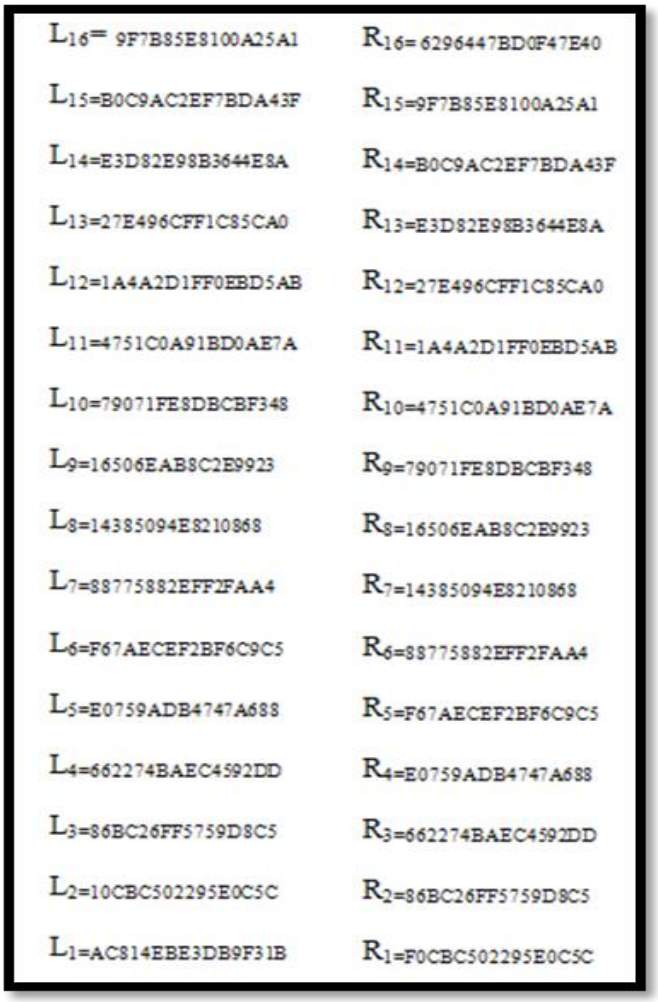

Figure 7: Output of the Decryption

Once all the steps for decryption have been properly done, the original plaintext will be returned. For the plaintext BA0FC5E7ABE90C79 is retrieved from the ciphertext of 94312EBA62C07654.

VI. CONCLUSION AND RECOMMENDATION

Based on the study on enhancing DES, which is to improve its security, the authors have come up with several conclusions as to why the new 128-bit DES that has been made is an improvement to the original. First of all, the original 64-bit blocks that were used as the basis for the standard DES has been doubled to its original size. This results in bigger blocks that leave more room for security due to the ability to expand upon the sheer size of it.

For example, many of the inner-workings of the DES have been remodeled and enlarged; the Initial Permutation, PC-1, and PC-2 tables that the standard DES used can now utilize the 128-bit size of the modified DES. Because of the expansion of these tables, brute force attacks should take approximately twice as long, considering the additional steps that would be needed to decrypt the modified block cipher.

Overall, the aim of the authors was to improve the security of DES against direct attacks. Due to the expanded size of the new DES, it is satisfactory to say that the original aim, which was to improve the defense of DES against brute force attacks, of which it was weak against, has been met.

The authors' suggestion for future researchers who are also studying DES, is adding more security measures to DES. One example to be considered is expanding the number of bits even further beyond 128-bits. This can be either 256-bits or 512- bits. Future researchers can also add their own ideas, such as implementing a time lock where the message can only be opened within a specific time span, or simply adding additional passwords to improve the security.

ACKNOWLEDGEMENT

The authors would like to express their humble gratitude towards their peers at the University of the East Caloocan for giving them the necessary strength to finish this paper, as well as their friends, families and loved ones. This would not be possible without the inspiration that they have given. Last but not least, the authors would like to thank God, for giving them spiritual guidance during the trying times that went into writing this research paper.

REFERENCES

[1] Pranav M , Archana K Rajan , *"DES security Enhancement with Dynamic Permutation"*, International Conference on Applied and Theoretical Computing and Communication Technology. 2015

[2] Putra S, Ahmad A, Sutikno S, *"Power Analysis Attack on Implementation of DES*", International Conference on Information Technology Systems and Innovation, Purdue University, 2016

[3] Kak A, *"Lecture 3 Block Ciphers and the Data Encryption Standard"*, 2017

[4] Mohamed A. Seif Eldeen, Abdelatif A. Elkouny, Salwa, Elramly *"DES Algorithm Security Fortification Using Elliptic Curve Cryptography"*, IEEE, 2015

[5] Davis, R., *"The Data Encryption Standard in Perspective,"* Proceeding of Communication Society magazine, IEEE, Volume 16 No 6, pp. 5-6, Nov. 1978.

[6] Mitali, Kumar, V., and Sharma, A., *"A Survey on Various Cryptography Techniques"* International Journal of Emerging Trends & Technology in Computer Science, Volume 3 Issue 4, 2014

[7] Rhee, M., *"Internet Security: Cryptographic Principles, Algorithms and Protocols"* Wiley. 2003

[8] Kenekayoro, P., *"The data encryption standard thirty four years later: An overview"* African Journal of Mathematics and Computer Science Research. 2010

[9] AL-Hamami, A., AL-Hamamim M. , Hashem S, *"A Proposed Modified Data Encryption Standard algorithm by Using Fusing Data Technique"*, World of Computer Science and Information Technology Journal Vol. 1, No. 3, 88-91, 2011

[10] Connell, C, *"An Analysis of NEWDES: A Modified Version of DES"*, CRYPTOLOGIA, 1990

[11] Stanisavljeviü, Ž, *"Data Encryption Standard Visual Representation,"* IEEE, 2015

[12] Ren, W. & Zhiqian, M. *"A Hybrid Encryption Algorithm Based on DES and RSA in Bluetooth Communication,"* Second International Conference on Modeling, Simulation and Visualization Methods, 2010

[13] Zibideh, W. Y. & Matalgah, M. M. *"Modified-DES encryption algorithm with improved BER performance in wirelesscommunication,"* Radio and Wireless Symposium (RWS) IEEE, 2011

[14] Sensarma, D & Sen Sarma, S. *"GMDES: A Graph Based Modified Data Encryption Standard Algorithm with Enhanced Security,"* International Journal of Research in Engineering and Technology Volume 3 Issue 3, 2014

[15] Shah, K. R. & Gambhava, B. *"New Approach of Data Encryption Standard Algorithm,"* International Journal of Soft Computing and Engineering, 2012

[16] Mani, K. & Devi, A. *"Modified DES using Different Keystreams Based on Primitive Pythagorean Triples"* I.J. Mathematical Sciences and Computing, 2017,

[17] Sharma, M., Garg, R. B. & Dwivedi, S. *"Comparative Analysis of NPN Algorithm & DES Algorithm"* IEEE, 2014

[18] Singh, G. & Supriya *"A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security."* International Journal of Computer Applications (0975 – 8887) Volume 67– No.19, 2013

[19] Schneier, B. *"A Self-Study Course in Block Cipher Analysis"*, CRYPTOLOGIA

[20] National Bureau of Standards *"Computer Security and the Data Encryption Standard"* NBS Special Publication 500-27. 1977

[21] Singh, A. K. & Varshney, S. *"Enhanced Data Encryption Standard using Variable Size Key (128N Bits) and 96 Bit Subkey*", International Journal of Computer Applications (0975 – 8887) Volume 98– No.8, 2014

[22] Grabbe, J. *"The DES Algorithm Illustrated*", retrieved from http://www.aci.net/Kalliste/des.htm

