TUDE Block Cipher

Pandyaka Aptanagi[1], **M. Rifky I. Bariansyah**[2].

[1,2] Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika (STEI), Institut Teknologi Bandung (ITB), Jalan Ganesha 10, Bandung 40132
E-mail: 13517003@std.stei.itb.ac.id, 13517081@std.stei.itb.ac.id

Abstract. Cryptography is an integral part of modern world information security. One of the important innovations in cryptography is block cipher. Many standards like DES and AES are leveraging this method to secure information exchange in everyday activity. In this paper we present TUDE block cipher, a block cipher experimentation that takes parts from DES and AES with several modifications. TUDE block cipher operates on 128 bits message and 64 bits key with a notable modification in its key generation and round function. Our observation shows that it is relatively secure from brute force attacks and able to withstand frequency analysis attacks since its compliance with Shannon's principle **Keywords**: cryptography, block cipher, diffusion, confusion, iterated cipher, Feistel network, S-box, DES, AES

1. Introduction

We exchange information every second of the day. However, not all information is for everybody. A piece of information is for someone who has the right and not anyone else. It has been hundreds of years of effort finding ways to protect information. Since Julius Caesar hid a message for his general in the war front until now. The method of protecting information from adversaries so only those intended can read it is called cryptography. One of the remarkable milestones in modern cryptography is block cipher.

DES is an example of block cipher. DES or Data Encryption Standard is a standard for a symmetric key block cipher. DES was created in 1972 and was based on the Lucifer algorithm by Horst Feistel. It was adopted by the National Bureau of Standards (NBS) as a standard encryption algorithm after its assessment by the National Security Agency (NSA). DES operates on 64 bit blocks and 64 bit key with 8 unused parity bits (56 bit is used)[2]. The block first will go through initial permutation, then it will be encrypted in a 16 times enciphering with different internal keys each, after that it will go through a final permutation (inverse initial permutation). A study by Mahajan and Sachdeva [1], shows the method is vulnerable to brute force, linear, and differential cryptanalysis attacks.

Another example is AES. AES or Advanced Encryption Standard is also a standard for a symmetric key block cipher. AES was meant to be a more secure replacement for DES. In October 2000 National Institute of Standards and Technology (NIST) chose Rijndael algorithm as the AES. Rijndael supports key length from 128 to 256 bit (multiple of 32) and is independent of block size. The algorithm works on bytes rather than bits. The Rijndael algorithm consists of multiple rounds, each containing substitution with S-box, wrapping, column mixing and adding a round key[3]. A study by Mahajan and Sachdeva [1], shows the method has superior security but is still vulnerable to brute force attacks.

In this paper we propose TUDE block cipher operates on 128 bits message and 64 bits key. Parts from DES and AES are used to implement this block cipher. We leverage substitution functions like

ROT13 and S-box, permutation using random sampling, and logical complement operation in the algorithm. The ultimate goal of this experimentation is to have a new block cipher that is secure and adheres to Shannon's ideas of secure cipher.

2. Literature Review

In this following chapter we will help readers to gain understanding by discussing relevant works and ideas.

2.1. Block Cipher

In Block Cipher, plaintext bits are divided into blocks of bits with the same length. The encryption will be done to each block. The ciphertext and plaintext length will be one and the same. Block Cipher has five modes of operation which are[4]:

2.1.1. Electronic Code Book (ECB)

Every plaintext block P_i will be encrypted individually, independently from each other, and yield ciphertext block C_i. With this mode, encryption does not have to be done sequentially and a mistake in a block won't affect others. However, because a plaintext may contain recurring parts, encryption may yield identical ciphertext blocks. This disadvantage makes the mode vulnerable to frequency analysis attacks.

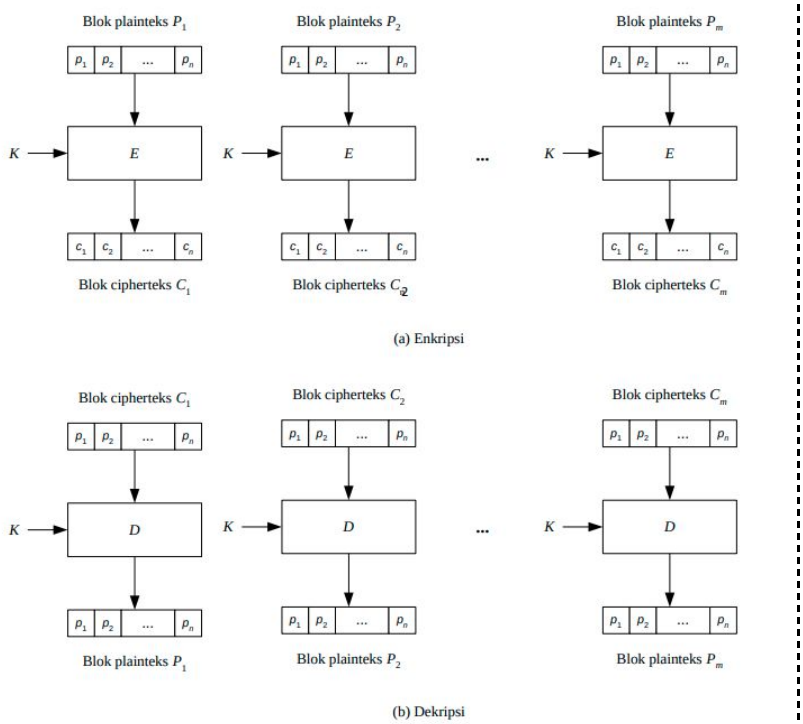

Figure 2.1. Example of Electronic Code Book (Source: http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Kripto-modern-2020-bagian3.pdf)

2.1.2. Cipher Block Chaining (CBC)

To overcome ECB disadvantages, CBC proposes the idea of creating dependence between blocks. The result from the previous block encryption will be received by the current block as a feedback. Using this approach, identical plaintext blocks won't yield the same cipher blocks. However, because of dependencies between blocks, one mistake in one block encryption will affect the subsequent blocks.

Regardless of the error caused in encryption, in decryption the error will only affect the corresponding block and one bit on the next block.

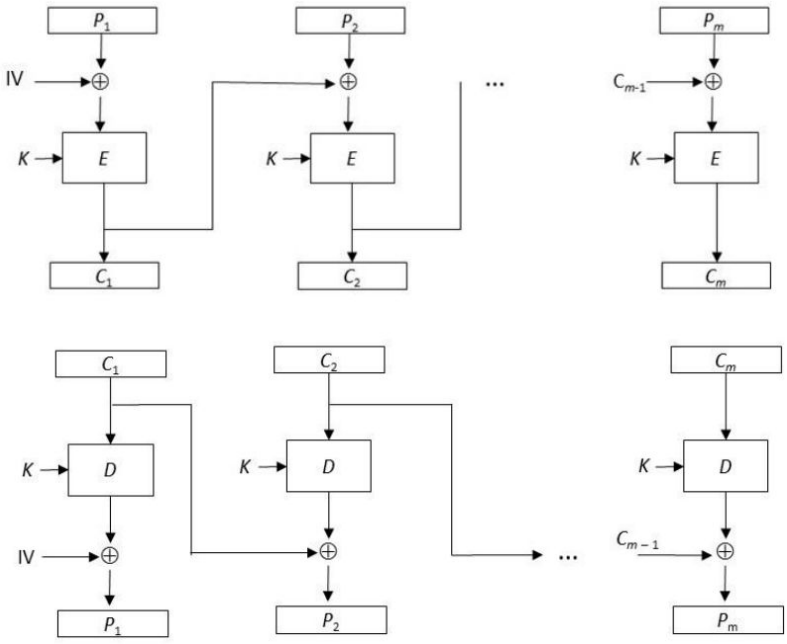

Figure 2.2. Example of Cipher Block Chaining (Source: http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Kripto-modern-2020-bagian3.pdf)

2.1.3. Cipher-Feedback (CFB)

Cipher-Feedback is created to provide encryption for message size smaller than one block size. CFB uses a queue with the size of the input block and will treat block cipher like stream cipher. It still has the same disadvantage as CBC regarding chain reaction towards error.

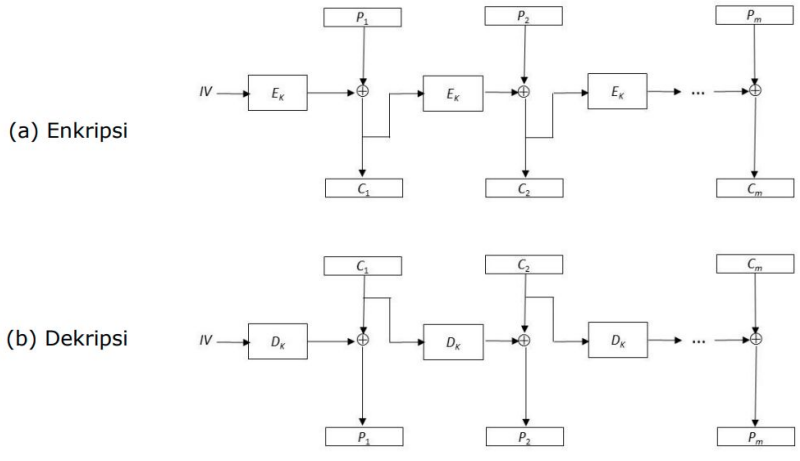

Figure 2.3. Example of Cipher Feedback Operation (Source: http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Kripto-modern-2020-bagian3.pdf)

2.1.4. Output-Feedback (OFB)

This mode is akin to CFB, except it sends the encrypted output as feedback instead of the actual ciphertext to the rightmost space in the queue. Using this approach, a one bit mistake will only affect the corresponding block both in encryption and decryption. OFB is suitable for digitized analog transmissions like video or audio, where one bit mistake is tolerable but propagated error is not.

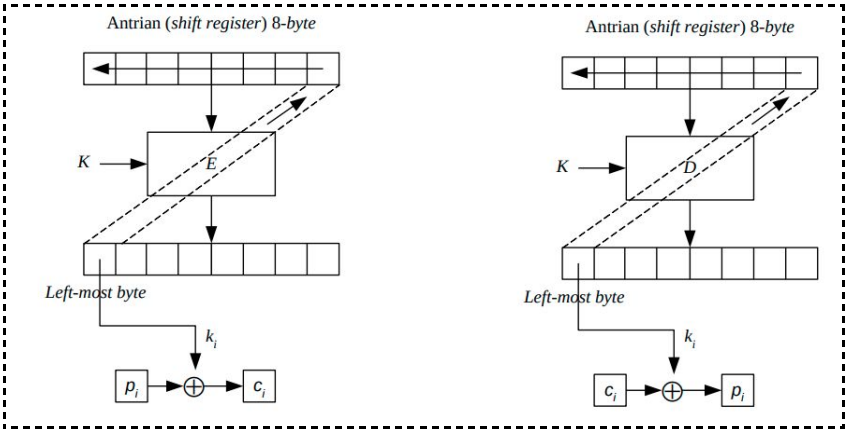

Figure 2.4. Example of Output Feedback Operation (Source: http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Kripto-modern-2020-bagian3.pdf)

2.1.5. Counter Mode

Counter mode does not have dependency between blocks. It has something called a counter which is a block of bits with the size of a plaintext block. Counter value must differ across blocks. The first step is initializing the counter with the same value, then for every block encryption, the counter value will be incremented by one.

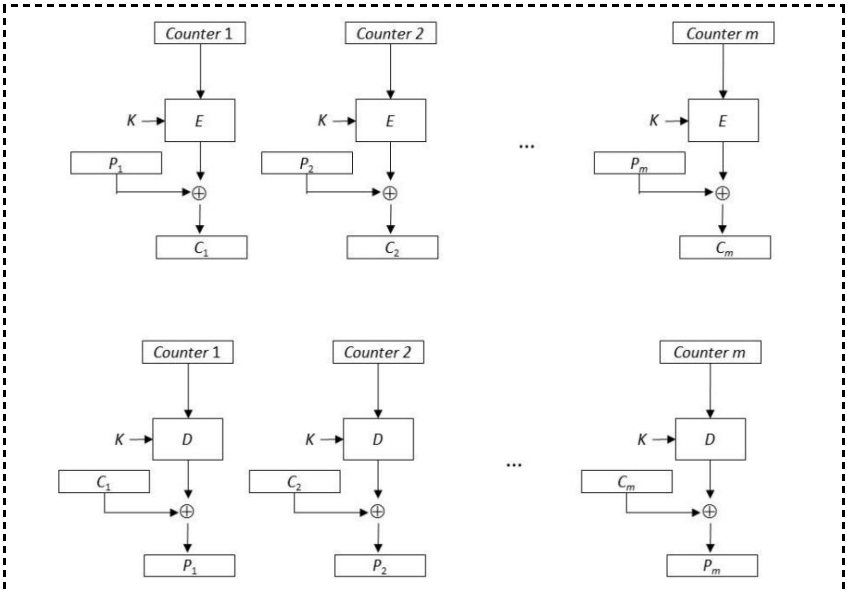

Figure 2.5. Example of Counter Mode Operation (Source: http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Kripto-modern-2020-bagian3.pdf)

2.2. Diffusion and Confusion

The DES, AES, and many more block ciphers are designed using two properties introduced by Claude Shannon in 1945, diffusion and confusion. The former idea is that each plaintext block bit or key bit affects many bits of the ciphertext block, CBC and CFB implement this idea. The latter idea is that each bit of the ciphertext block has high nonlinear relations with the plaintext block bits and the key bits, a complex substitution algorithm can be used to implement this idea[5][6]. The encryption and decryption functions of a cipher should have both good diffusion and confusion for the message block bits and key bits.

2.3. Iterated Cipher

Iterated cipher is a simple transformation function that transforms plaintext into ciphertext several times[5]. Every round will be using a subkey or a round key combined with plaintext. Iterated cipher is expressed as equation (1)

$$C_i = f(C_{i-1}, K_i) \tag{1}$$

2.4. Feistel Network

Feistel Network is one of the methods of constructing block cipher. Algorithms like DES, LOKI, GOST, FEAL, Lucifer and Blowfish are using Feistel Network. This method is reversible, which means we don't need to make another algorithm for decryption.

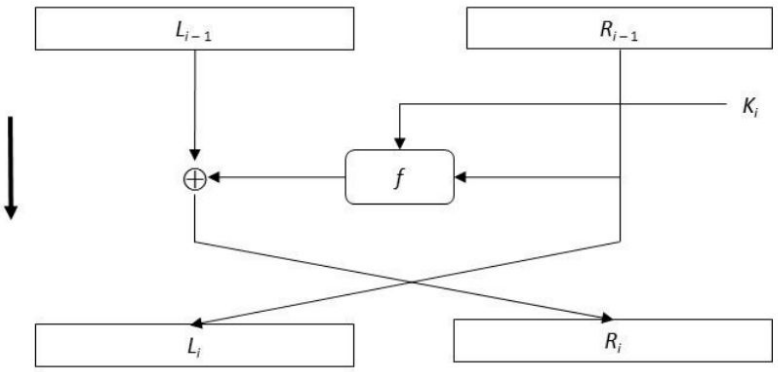

Figure 2.6. Example of Feistel Network Encryption (Source: http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Kripto-modern-2020-bagian4.pdf)

Figure 2.6 can also be expressed as equation (2) and (3)

$$L_i = R_{i-1} \tag{2}$$

$$R_i = L_{i-1} \otimes f(R_{i-1}, K_i) \tag{3}$$

2.5. S-box

S-box is a matrix of substitutes that maps one or more bits to another one or more bits. An S-box that maps m bit of input to n bit of of output is called $m \times n$ S-box. This step is using a look-up table operation, hence it is non-linear[5].

3. Proposed Block Cipher

In this paper we propose TUDE block cipher that operates with 128 bit blocks and 64 bit keys on ECB, CBC, and Counter mode. However, users can input the key with any length. Every block will go through 16 rounds of encryption with different keys each. This chapter will discuss the implementation of the block cipher. The encryption process consists of:

1. Converting text into blocks of messages
2. Generate keys for each round
3. Encrypt each block through feistel network
4. Convert blocks back to string

The decryption process has a similar process like encryption except it has a different sequence in the feistel network. Some notable modifications in the algorithm are in the key generation and the round function used in the feistel network.

3.1. Key Generation

In this step we make use of shift substitution, permutation, and bit shifting. The complete steps are as follows:

1. Shift substitution using ROT13, ROT13 is a function that substitutes a letter with the 13th letter after it in the alphabet
2. Transform substituted string into bits
3. For each round, from the bits we take 64 bit permutation as a round key. We leverage random function from python to take 64 random samples from the bits
4. Shift the round key to the left four times

Finally we will have an array of 16 round keys.

3.2. Round Function

Every block will go through 16 rounds, each round contains:

1. Firstly, we carry out an XOR operation between the block and round key
2. Transform the result from the first step from bit to hexadecimal
3. To fulfill Shannon's confusion property, we leverage Rijndael S-box (AES) to perform substitution. The S-box is illustrated in figure 3.1.
4. Transform the result from the third step from hexadecimal back to bit
5. To fulfill Shannon's diffusion property, we carry out permutation by grouping numbers with odd indexes and even indexes. For example [0 1 2 3 4 5] will be grouped into [0 2 4 1 3 5].
6. Then, we carry out logical complement operation or negation to the result from step 5. For example [0 1 1 0 1] will be negated to [1 0 0 1 0]

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Figure 3.1. Rijndael S-Box for Advanced Encryption Standard (Source: http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Review-beberapa-block-cipher-dan-stream-cipher-2020-bagian4.pdf)

4. Experiment and Evaluation

In this chapter we will show and evaluate the experiment of the TUDE block cipher. We conduct the experiment with different plaintext sizes across ECB, CBC, and counter mode.

4.1. Experiment

The following are experimentation results for each operation mode with a 128 bytes plaintext.

Plaintext	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
Key	ini adalah kunci
Ciphertext	#=;§?µÿx¬>vØý;Å,Ô°i@J\+y<eR!ÊåÍ»ó':«=%I @7 Æ´_Þç,@©½gµ&l) n,×5Ôlc³Tl|j»#3}tsn ¹`Â©4]W¯@ÆðA¬°¦JZ\ÿº

Table 4.1. Experiment Result using ECB Mode

Plaintext	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
Key	bariansyah
Ciphertext	¨pp× ¿ÕW,oÀ£~Êú?Å`_Zï/÷åPùeåù¡·Þ¼më»ÂIÅêÞ5 ß× 5ÜÉ<Xp|ºCss',4/ÇdPpeq,¨Å(Èè¤ÙsÂå¶þ°;Ë½

Table 4.2. Experiment Result using CBC Mode

Plaintext	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
Key	pandyaka
Ciphertext	ÑT`à¸[¡H|cz³CZ¶ ?ÀBRÝ®ë¦¢

	þ¤ëH· ±d3¶5¥e©,´q±j^e¢S eBîØýî^KàAl bÏ)6¬KÛ×Ò1Òì¥ÇH}êl×YMç,îTÀC@

Table 4.3. Experiment Result using Counter Mode

Time comparisons between each mode are shown in Table 4.4. where we can see insignificant differences among them.

Mode	Operation	Message Size	
		128 bytes (s)	12800 bytes (s)
ECB	Encrypt	0.00717926025390625	0.5345511436462402
	Decrypt	0.00746583938598632 8	0.5345840454101562
CBC	Encrypt	0.00720953941345214 84	0.5330193042755127
	Decrypt	0.00689506530761718 75	0.5586462020874023
Counter	Encrypt	0.00727820396423339 84	0.5266101360321045
	Decrypt	0.00940632820129394 5	0.5391757488250732

Table 4.4. Execution Time Comparison Table

4.2. Evaluation

In this subchapter we will discuss brute force attacks against TUDE block cipher and it's compliance regarding Shannon's confusion and diffusion.

4.2.1. Brute Force Analysis

Brute Force is a trial-and-error attack where the attacker generates all possible combinations of keys to decrypt an encrypted message. In sometime Brute Force will obviously guess a correct combination, however it takes a vast amount of time to find a key in an impossibly large prospect. In theory the larger the size of the key, the harder it is to find. TUDE block cipher uses a 64 bit sized key. This means the possible number of possible key combinations are 2^{64} or $1.8446744073709552 \times 10^{19}$. If an attacker uses a machine that can generate 10^6 keys a second, they will need $1.8446744073709552 \times 10^{13}$ seconds or 584,942.417355068 years to generate all possible keys. From this observation, we can see that TUDE block cipher is relatively safe from Brute Force attacks.

4.2.2. Confusion and Diffusion Analysis

We observed the confusion property of TUDE block cipher by doing a frequency analysis to see the distribution of characters in plaintext and ciphertext with the size of 12800 bytes. The observation histogram is illustrated in figure 4.1.

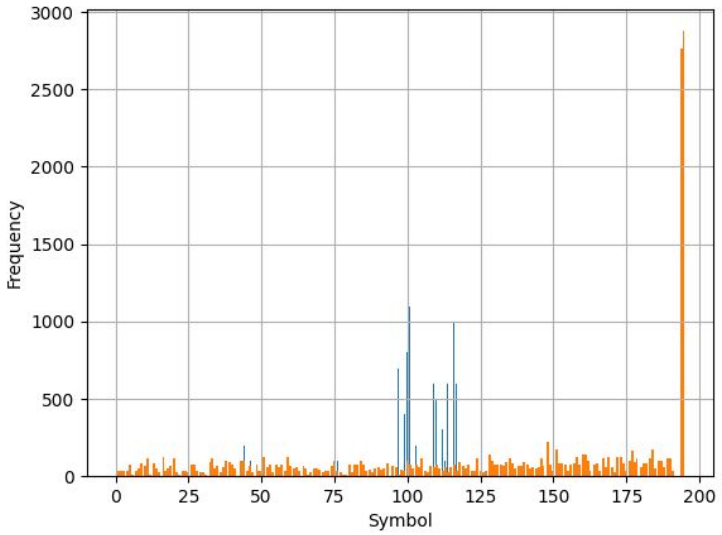

Figure 4.1. Frequency Analysis Histogram (blue for plaintext, orange for ciphertext)

As shown in figure 4.1., ciphertext from TUDE block cipher has a uniform character distribution. That means each bit of the ciphertext block has high nonlinear relations with the plaintext block bits and the key bits, we leverage substitution to implement this idea. It shows that TUDE block cipher has implemented Shannon's confusion idea correctly.

We observed the diffusion property of TUDE block cipher by doing a comparison analysis between ciphertext with modification in keys. Table 4.5. shows encryption with keys that have a single character difference. We can see that the ciphertext are completely different and show no similarity in any way. Table 4.6. shows decryption with keys that have a single character difference (one is correct). We can see that using a slightly modified key, i.e. one character difference, will yield a completely different result. The observation manifests that a small difference in a key affects many bits of the ciphertext block. It shows that TUDE block cipher has implemented Shannon's diffusion idea correctly.

Plaintext	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
Key 1	ini adalah kunci
Key 2	ini adalah konci
Ciphertext 1	#=;§?µÿx¬>vØý;Å,Ô°i@J\+y<eR!ÊåÍ»ó':«=%I@7 Æ´_Þç¸@©½gµ&l) n¸×5Ôlc³Tl|j»#3}tsn ¹`Â©4]W¯@ÆðA¬°¦JZ\ÿº
Ciphertext 2	ÐÐht<Î;¨RìNt¿(e¤á)t«SÒºq{´wOa[N5Ê[6Á¸M« >¥ ÅòO7ó]ß½ Hb/æ/1ð An?ùâ3C6zG7´áÙnï µ¦éÇ-v

Table 4.5. Encryption with Different Key

Ciphertext	#=;§?µÿx¬>vØý;Å,Ô°i@J\+y<eR!ÊåÍ»ó':«=%I@7 Æ´_Þç¸@©½gµ&l) n¸×5Ôlc³Tl|j»#3}tsn ¹`Â©4]W¯@ÆðA¬°¦JZ\ÿº
Key 1	ini adalah kunci
Key 2	ini adalah konci
Plaintext 1	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
Plaintext 2	\Io4ÌeÍøH°ÊN {¨ê ïç®éî§Ug¨|½S¾&Aà S÷rqRØ|æ,ð¾ÒÂÕM5¯Ý U½g¢¸«c0Wtg|¼çwÌ TUëqÂ5ÚDE~w7[¬Ù%£^W¶þÉù%

Table 4.6. Decryption with Different Key

5. Conclusion and Future Work

TUDE block cipher is one of block cipher variants that has good performance based on experiment result and evaluation. Because of its simple key generation and round function, TUDE block cipher has a relatively fast execution time and can be implemented on low computational power machines. This block cipher is relatively safe from Brute Force attacks. It is also able to withstand frequency analysis attacks since its implementation of Shannon's diffusion and confusion principle.

For future work, security performance and computational time can be improved by having a dynamic block size and plaintext-based key generation.

6. References

[1] Mahajan, P., & Sachdeva, A. (2013). A study of encryption algorithms AES, DES and RSA for security. *Global Journal of Computer Science and Technology*.

[2] Munir, R. (2020). *Review Beberapa Block Cipher dan Stream Cipher (Bagian 2: DES)*. IF4020 Kriptografi Informatics Engineering ITB, Bandung.

[3] Munir, R. (2020). *Review Beberapa Block Cipher dan Stream Cipher (Bagian 4: AES)*. IF4020 Kriptografi Informatics Engineering ITB, Bandung.

[4] Munir, R. (2020). *Kriptografi Modern (Bagian 3: Block Cipher)*. IF4020 Kriptografi Informatics Engineering ITB, Bandung

[5] Munir, R. (2020). *Kriptografi Modern (Bagian 4: Perancangan Block Cipher)*. IF4020 Kriptografi Informatics Engineering ITB, Bandung

[6] Ding, C. (2020). *Shannon's Idea of Confusion and Diffusion*. CSIT5710: Cryptography and Security Computer Science UST, Hong Kong.

Acknowledgments

Thank God for his blessings we are able to finish this project with great studiousness. We would like to thank our family and friends for supporting us in the making of this paper. We would also like to thank Dr. Ir. Rinaldi Munir for his guidance and kindness. Here's to the next half semester.