Double Chaining Algorithm

A Secure Symmetric-key Encryption Algorithm

Daniar Heri Kurniawan
Department of Informatics
Institute Technology, Bandung
Bandung, Indonesia
daniar.h.k@gmail.com

Rinaldi Munir
Department of Informatics
Institute Technology, Bandung
Bandung, Indonesia
rinaldi@informatika.org

Abstract— **Technology is a key of innovation in any aspect of this modern era. In every technology, data becomes the most important asset to be protected. Many encryption algorithms are widely available and used in information security. Encryption can provide secure information across platform. Encryption algorithms are classified into two groups: symmetric-key (also called secret-key) and asymmetric-key (also called public-key). Generally, asymmetric-key encryption is used along side symmetric-key encryption to get the best performance of data transfer. Earlier many researchers have proposed various encryption algorithms such as AES, DES, Blowfish, etc. However, as security level is increasing, the time and complexity of algorithm is also increasing. This is the major cause of decreasing the speed and efficiency of the encryption system. In this paper we have proposed a new encryption algorithm "Double Chaining Algorithm (DCA)" which enhances the security performance. The DCA is a symmetric-key encryption algorithm that uses 128-256 bits key size. This algorithm uses XOR operation for the chaining process so that it has small complexity and can be implemented easily in hardware mode to increase its encryption process. This paper is an attempt to invent a new encryption model which is secure and very fast. The average Avalanche Effect in this algorithm is 0.996 out of 1.00. Moreover, the experiment and analysis result gives further proof of the DCA's strength.**

Keywords— symmetric-key encryption; information security Double Chaining Algorithm; Avalanche Effect

I. INTRODUCTION

Technology is a key of innovation in any aspect of this modern era. In every technology, data becomes the most important asset to be protected. Such transactions (over wire or wireless public networks demand end-to-end secure connections) should be confidential, to ensure data authentication, accountability and confidentiality, integrity and availability, also known as CIA triad [1]. Many encryption algorithms are widely available and used in information security. Encryption algorithms are classified into two groups: symmetric-key (also called secret-key) and asymmetric-key (also called public-key) encryption. Symmetric-key encryption is a form of cryptosystem in which encryption and decryption are performed using the same key. It is also known as conventional encryption. Asymmetric-key encryption is a form of cryptosystem in which encryption and decryption are performed using the different keys – one a public key and one a private key. It is also known as public-key encryption [3].

A key is a numeric or alpha-numeric text or may be a special symbol. The key is used at the time of encryption takes place on a plaintext and at the time of decryption takes place on a cipher text. The selection of key in Cryptography is very important since the security of encryption algorithm depends directly on it. The strength of the encryption algorithm relies on the secrecy of the key, length of the key, the initialization vector, and how they all work together [4]. Asymmetric-key encryption is about 1000 times slower than symmetric-key encryption which makes it impractical when trying to encrypt large amounts of data. Also to get the same security strength as symmetric, asymmetric must use a stronger key than symmetric encryption technique [5].

Generally, asymmetric-key encryption is used along side symmetric-key encryption to get the best performance of data transfer. Since asymmetric keys are bigger than symmetric keys, asymmetrically encrypted data is tougher to crack than symmetrically encrypted data. However, this does not mean that asymmetric keys are better. Rather than being compared by their size, these keys should compared by the following properties: computational burden and ease of distribution.

In this research paper, a new encryption algorithm named **"Double Chaining Algorithm (DCA)"** is proposed which is applying block chaining as its core process. The DCA is a symmetric-key encryption algorithm that uses 128 - 256 bits key size. There are two different size of block and two different level of process. However, this algorithm has small complexity since it implements identic process for any size of block. The most common operation in DCA is XOR calculation which can be implemented easily in hardware mode as well as in software mode to ease the distribution.

The rest of this paper is organized as follows: Section 2 surveys some well-known encryption algorithm. We discuss some characteristics of the encryption algorithm and the criteria for evaluating the security of the algorithm in Section 3. Section 4 describes the basic operations of the DCA. Next, we propose our new encryption algorithm in Section 5. The experiment result and analysis of the proposed algorithm are presented and discussed in Section 6, 7, and 8. Finally, Section 9 concludes this paper.

978-1-5090-1636-5/16/$31.00 ©2016 IEEE

II. Overview The Existing Algorithm

Encryption is a well known technology for protecting sensitive data. These are the existing algorithm that widely applied in various technology:

A. Data Encryption Standard (DES)

Data Encryption Standard (DES) algorithm's purpose is to provide a standard method for protecting sensitive commercial and unclassified data. It uses a 56-bit length key and it can be implemented in hardware and software by executing the algorithm 16 times, although the key is considered un-secure because of the length. There are many attacks and methods to exploit the weaknesses of DES recorded until now, which made it an insecure block cipher. The 3-DES is an improvement over the DES because it employs the original DES 3 times and it uses a 192-bit length key, which makes the computation time too long and not suitable for RTA (Real Time Application) [6].

B. Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) algorithm is not only used for security but also for great speed purpose. This algorithm is an encryption standard recommended by NIST (National Institute of Standards and Technology) to replace DES. Encrypts data blocks of 128 bits in 10, 12 and 14 round depends on its key size. It can be implemented on various platforms especially in small devices and already carefully tested for many security applications. The AES algorithm performs four steps (bytes substitution, shift rows, mix columns, and add round key) on each block of 128-bit plaintext in every round. If a 128-bit key is used it performs 10 rounds, if 192-bit key is used it performs 12 rounds and if a 256-bit key used it performs 14 rounds, which makes this protocol ideal for RTA encryption/decryption like voice and signaling [7].

C. Rivest-Shamir-Adleman (RSA)

The most widely used Public-Key algorithm is RSA. It is an asymmetric (public key) cryptosystem based on number theory, which is a block cipher system. It uses two prime numbers to generate the public and private keys. These two different keys are used for encryption and decryption purpose. RSA algorithm involves these steps: key generation, encryption, and decryption. The RSA is an exponentiation cipher which is very popular in business applications. Moreover, the RSA is built in operating systems by Microsoft, Apple, Sun and Novell. It is also found in secure telephones, Ethernet Network cards and on smart cards [7].

III. The Characteristic of Secure Encryption

Claude Shannon introduced an idea of substitution-permutation (S-P) networks in 1949 – the basis of modern block ciphers [9]. There are some general explanation about characteristics of good ciphers by Claude Shannon [10], which are:

1. The amount of secrecy needed should determine the amount of labor appropriate for the encryption and decryption
2. The set of keys and the enciphering algorithm should be free from complexity
3. The implementation of the process should be as simple as possible
4. Errors in ciphering should not propagate and cause corruption of further information in the message
5. The size of the enciphered text should be no larger then the text of the original message.

Moreover, S-P networks are based on two primitive cryptographic operations: substitution (S-box) and permutation (P-box). Both of the operation will provide confusion and diffusion of message. Confusion means that the process drastically changes data from the input to the output. The interceptor should not be able to predict what changing one character in the plaintext will do to the cipher text. An algorithm providing good confusion will have a complex functional relationship between the plaintext, key pair and the cipher text.

Diffusion means that changing a single character of the input will change many characters of the output. A good diffusion will make every part of the input affects to every part of the output, which make analysis become much harder. No confusion process is perfect: it always lets through some patterns. Good diffusion scatters those patterns widely through the output. This makes patterns vastly harder to spot, and vastly increases the amount of data to analyze to break the cipher.

In cryptography, the Avalanche Effect that refers to a desirable property of cryptographic algorithms, typically block ciphers and cryptographic hash functions. The Avalanche Effect is evident if an input is changed slightly (for example, flipping a single bit), then the output changes significantly (e.g., half the output bits flipped). In the case of high-quality block ciphers, such a small change in either the key or the plaintext should cause a drastic change in the cipher text. The actual term was first used by Horst Feistel [11], although the concept dates back to at least Shannon's diffusion. The SHA-1 hash function exhibits good Avalanche Effect. When a single bit is changed the hash sum becomes completely different.

$$\text{Avalanche Effect} = \frac{\text{Number of flipped bits in ciphered text}}{\text{Number of bits in ciphered text}}$$

Figure III-1 The Equation of Avalanche Effect

IV. The Basic Operation

A. SHA-256

The Secure Hash Algorithm (SHA) is a set of cryptographic hash functions, which includes SHA-224, SHA-256, and SHA-512. A cryptographic hash is like a signature for a text or a data file. The 256 in SHA-256 represents the bit size of the hash output or digest when the hash function is performed. SHA-256 algorithm generates an almost-unique, fixed size 256-bit (32-byte) hash. Hash is a one way function, once it encrypted it cannot be decrypted back. This makes SHA suitable for password validation, challenge hash authentication, anti-tamper, and digital signatures.

B. *Binary Exclusive-or (XOR)*

Binary XOR operation (also known as binary XOR function) will always produce '1' as the output if either of its inputs is '1' and will produce '0' as the output if both of its inputs are '0' or '1'. The symbol of this operation is "⊕". This operation can be implemented in hardware easily because it is one of the basic binary operation.

C. *Chaining Operation*

Chaining Operation is an operation which takes two blocks of bits and then calculate their XOR result. The result will be used to replace the second block and to calculate the next XOR result with the third block. Those operations will be continued until there is no block to be replaced anymore. The idea of this operation is similar to iterated block ciphers with the XOR function as the round function. However, this algorithm will perform this operation in a special order so it can be differed from any existing encryption algorithm.

V. DOUBLE CHAINING ALGORITHM

Double Chaining Algorithm (DCA) is a block cipher encryption algorithm that uses 16 Bytes block length. It takes a block of plaintext bits and generates a block of cipher text bits, generally of same size. The encryption process that will be described in the next section maps n-bit plaintext blocks to n-bit cipher text blocks; n is called block length. It may be viewed as a simple substitution cipher with large character size. When the length of the data cannot be divided evenly by the number of 16 Bytes, the last block of bits needs to be padded up with redundant information (DCA uses null character for padding) so that the length of final block equals to block size of the scheme. The process of adding bits to the last block is referred as padding.

The DCA implements double Cipher-blok Chaining (double CBC) mode. A single CBC process is depicted on the Figure V-1. The DCA will do CBC process in two stages. The first stage is performed as a normal CBC mode (from the first block until the last block). Supposed that Cipher1 is the result of the first CBC, then the second stage of CBC is started in reversed order from the last block to the first block of Cipher1. Finally, the second CBC will result the cipher text of DCA. Each CBC stage will be repeated four times with different subkey for each attempt. Therefore, there are eight time of encrypting a single block using $\boldsymbol{E_k}$ function.

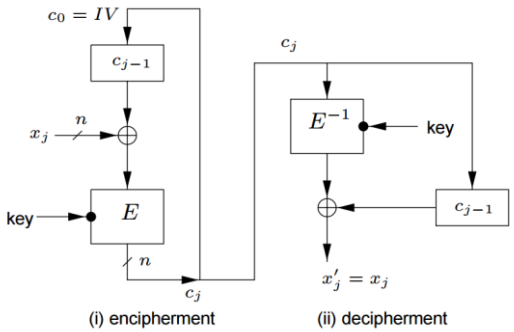

Figure V-1 Cipher-block Chaining Process

The key length is start from 8 Bytes to 32 Bytes. The SHA-256 hash function is applied to the key to get 256 bits output or digest. The digest will be used to generate subkey with total 4 subkeys. The number of round in DCA is 4 rounds. Each round consists of 2 process that will be described in the encryption section. The process of DCA is depicted below.

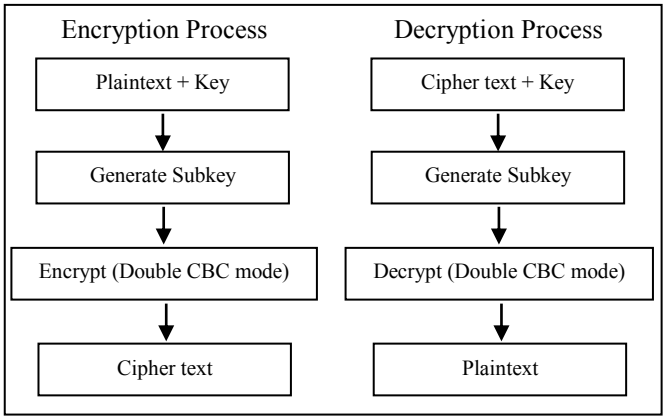

Figure V-2 The Decryption and Encryption Process of DCA

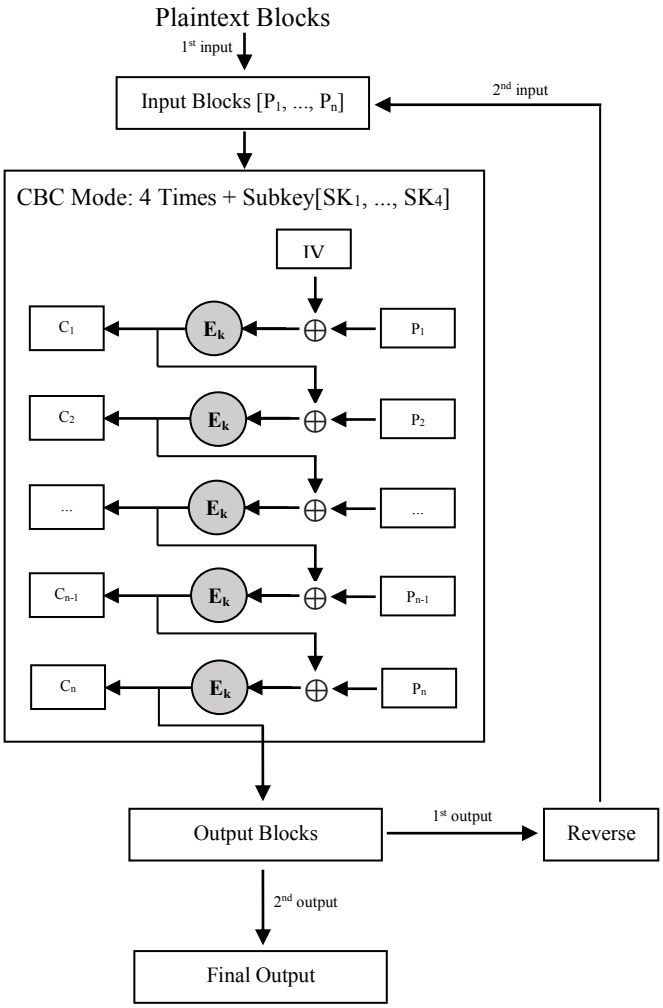

Figure V-3 Double CBC

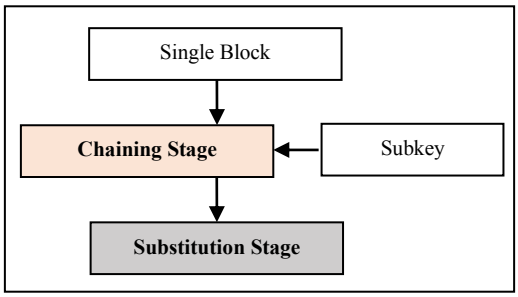

Figure V-5 The E_k Function

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

Figure V-7 Substitution Box

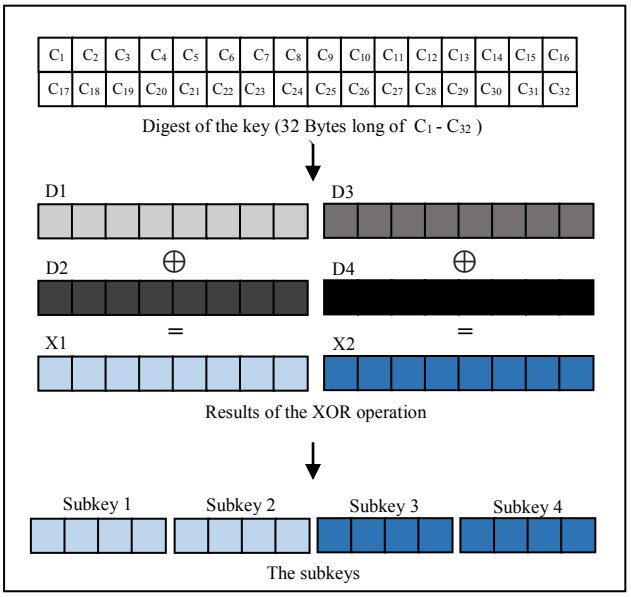

Figure V-6 The Subkey Generation Schema

A. *Subkey Generation*

During the encryption and decryption process, DCA uses different key for chaining process that will be described later. There are four rounds of chaining that require four new different keys. The new keys that are generated from original key are called subkeys. By finishing this stage, there are total four subkeys which are consist of two 4-Bytes long subkeys. The subkeys are generated based on the SHA-256 digest of the key. The steps are described as follows:

1. Compute SHA-256 of the Key (the output is 32 Bytes).
2. Supposed that the digest is split into 4 blocks of 8 Bytes: D1, D2, D3, D4. Calculate X1 and X2 as the XOR result of D1 ⊕ D2 and D3 ⊕ D4.
3. Finally, split X1 and X2 to the group of 4 Bytes which will be used as the subkeys.

B. *Encryption Process*

This process is done in two stage, which are chaining stage and substitution stage. By referring to Figure V-3, this process represents symbol $\boldsymbol{E_k}$. The input of this process is a single block and a subkey. The subkey is 4-Bytes long. It will be used in chaining stage. Both of the stages will be explained as follows:

- *Chaining Stage*

There are four rounds on this stage. Each of the round will use one Byte subkey. The 1-Byte key will be used for initiating XOR operation in every byte of the block (one block consists of 16 Byte) similar to stream cipher. For every byte of subkey, it will be used twice. The first is by running the 1-Byte stream cipher in a normal order, and the second is run in the reversed order. The Figure V-8 gives the more detailed explanation.

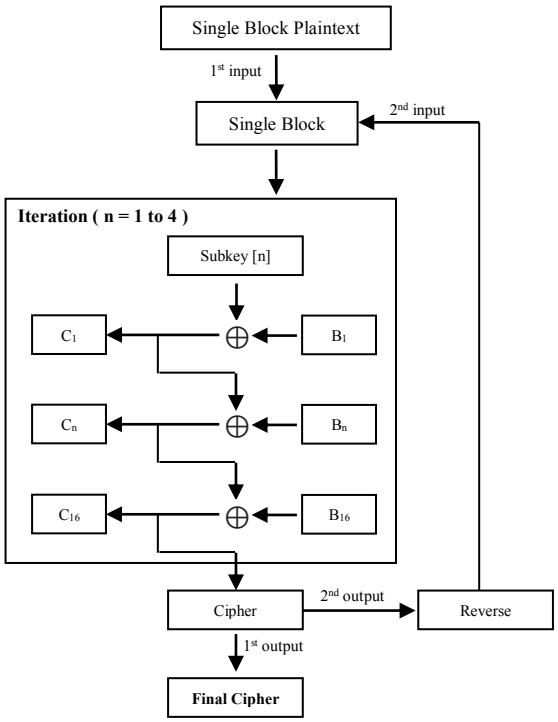

Figure V-8 The Chaining Stage

- *Substitution Stage*

This stage converts each byte of plaintext to cipher text according to S-Box (substitution table) in Figure V-7. The S-Box is static (constant) to ease of implementation in hardware

mode and to increase the performance of DCA algorithm. The S-Box consists of 256 bytes. The Figure V-7 is representing each byte in the form of hexadecimal. The x and y axis represent the index order.

C. Decryption Process

The decryption process is a reverse process of the encryption. By referring to the Figure V-3, the E_k function is changed to D_k function. The rest of the process can be derived from the encryption mechanism. In addition, the substitution box should be inverted for the substitution stage.

VI. Experiment Result

The DCA is implemented and tested in Java Language ver. 8. There is no special reason for choosing the language. The experiment is done for analyzing the Avalanche Effect (AE). The following table presents the encryption results of two plaintexts that is differ by one bit.

TABLE I. ENCRYPTION TEST 1 (SIMILAR PLAINTEKS)

Key 1	*hanubrh**f***
Plaintext 1	Lorem ipsum dolor sit amet, consectetur adipiscing **e**lit. Ipse Epicurus fortasse redderet, ut Sextus Peducaeus, Se.
Plaintext 2	Lorem ipsum dolor sit amet, consectetur adipiscing **f**lit. Ipse Epicurus fortasse redderet, ut Sextus Peducaeus, Se.
Cipher text 1	29c5 0326 d76a 3558 5198 2aca 2c19 95b2 62fb b93c 8b0e f369 2579 6173 b0ca 5d3b 263c ad6e c165 413e cea2 26f6 cd58 5b48 c2f2 efc7 3484 028e 2b5a b974 a902 53cd 32b2 a12e b1da 1390 255a 45d2 d3ce 0a27 7e3f 2ea2 b399 159f 1169 7dcf 693f 46e3 279e 336d cc6a 8c02 5438 fe74 75a7 abf7 79ab 2e62 4765 44c1 8fa9 9e2a 42cd 3fcc
Cipher text 2	c1db 25bb 87d1 6e7c c64a ffbd 4572 e9c9 98b8 58ae 6614 e5b3 3178 cb48 7dd6 c918 a80c 5b49 5525 e84b 4271 d500 4eef 3490 7a99 2339 b582 5d7a 2f7c f615 bf5b 2002 c6e9 807d f9ab 063e 76bb 34e0 f4da e92d 5f57 d57a 4629 43c5 50ea 7e0c 7400 0bf8 b3b9 f742 b087 951d 57a6 3c5e 4a6d e2bb c7d5 35a6 432f 7381 bc5b 22e1 b865 68d8
AE score	0.996

The next table is to test the significant changing in cipher text if a different key (differ by one bit) applied to the same plaintext. This will use the same plaintext as plaintext 1.

TABLE II. ENCRYPTION TEST 2 (SIMILAR KEY)

Key 2	*hanubrh**g***
Cipher text 3	b7e6 623b 4652 b15a 146b a1b5 b437 ee7a 44e1 9dec cf0b 8907 5979 c636 a1b0 f4e6 0d5a 2e82 2807 fa8f 8674 d3d9 1566 03b3 5fcc c61a 7e27 06fd f528 f64f cd63 0165 5c06 7c5e fdd4 53e0 c532 955b 30a2 91e5 c82e 066f 591d 2b63 f691 0290 48b1 ccc2 7cbc cf0d 0a7d f5f0 a5e4 a7c4 b2de 796f 7f79 2178 9460 2122 1bb5 54fc a7d0 72df
AE score	0.996

Both of the table show that the Avalanche Effect for the following test case is almost maximum. However, it cannot be generalized to any test case since the previous test case is only 115 Bytes characters length. Furthermore, this algorithm already tested to encrypt various text (English, Indonesian, Javanese, etc.) that is 0.5 - 5 MB length. The average **Avalanche Effect** is still **0.996.**

VII. Performance Analysis

In order to analyze its performance, the DCA algorithm is compared to other algorithm. There are three well known algorithm that we are comparing to, which are: Blowfish, 3-DES, and AES. We analyze the Avalanche Effect in each algorithm based on the various condition. The result are shown in the table below.

TABLE III. AVALANCHE EFFECT ANALYSIS

Plaintext's Modification Position	**Avalanche Effect**			
	Blowfish	**3-DES**	**AES**	**DCA**
In the begining	0.0002	0.91835	0.91835	**0.99606**
In the middle	0.0001	0.5034	0.50339	**0.99607**
In the end	0.0001	0.01673	0.01674	**0.99605**

The plaintext (King James Bible) that we use to calculate Avalanche Effect are differed by one bit. According to the result, the DCA always gets the highest Avalanche Effect. It proved that DCA is more secure than the other algorithm.

VIII. Security Analysis

The security analysis refers to common cryptanalyst attack, which are brute force attack, frequency analysis, and known plaintext attack.

A. Brute Force Attact

The time needed to find the right key using brute force attack depends on the length of the key. Since the range of the key's length is 128-256 bits, there will be $2^{128} + 2^{129} + ... + 2^{255} + 2^{256}$ possibilities or 2.31×10^{77} possibilities in total.

TABLE IV. THE TIME NEEDED FOR BRUTE FORCE ATTACK

Computer Performance	**Time Needed**
10^3 key / second	7.34×10^{66} years
10^6 key / second	7.34×10^{63} years
10^9 key / second	7.34×10^{60} years
10^{12} key / second	7.34×10^{57} years

B. Frequency Analysis Attack

In these following figures, the first chart shows the frequency of characters in plaintext (English text) and the second chart shows the frequency of characters in the cipher text. The x axis shows the byte of single character and the y axis shows the occurrence of every character.

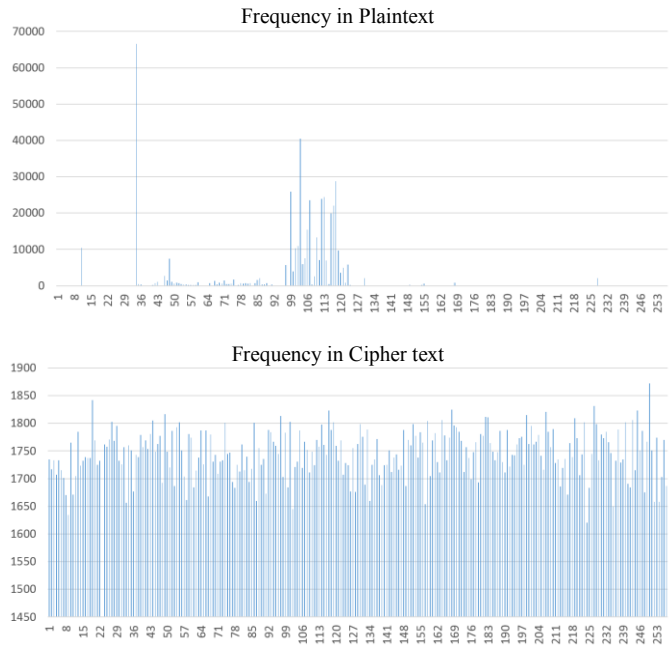

Figure VIII-1 Frequency of Every Character

In the plaintext, the highest character occurrence is in space character and on another character that often used in English language, such as E, T, and A. In another hand, the frequency of character's occurrence is almost equal. The lowest occurrence is 1620 times and the highest is 1872 times. By having standard deviation score in 42.64, it proofs that frequency analysis attack is very difficult.

C. *Known Plaintext Attack*

This attack needs a deep analysis for every cipher block. The security of DCA is determined by the complexity of its chaining process. That process will use different subkey for each attempt. Since the subkey is generated based on password's SHA-256 digest, it makes the attack become so difficult to achieve. Moreover, the length of the key will determine the strength of DCA the most.

IX. CONCLUSSION

In this paper, we proposed a new symmetric-key encryption algorithm named Double Chaining Algorithm (DCA). This algorithm was inspired by the importance of block cipher implementation in any modern cryptosystem. This algorithm is designed to ease the implementation in both, hardware and software mode. The computation that mostly used by this algorithm is XOR operation. There are two main stages for enciphering process, those are chaining stage and substitution stage. The key that is used in the chaining stage is derived from SHA-256 digest of the key. There are four subkeys that are generated and each of the key will be used separately for every iteration as described before. Moreover, this algorithm is already tested and implemented in Java 8 environment for the further analysis. The result of the analysis shows the power of DCA to face the brute force attack, frequency analysis attack, and known plaintext attack. The average Avalanche Effect in DCA is 0.996. However, there is no complex operation to achieve its high Avalanche Effect as described in the proposed algorithm section. Therefore this algorithm is secure and efficient to be implemented in both, software and hardware platform.

ACKNOWLEDGMENT

The authors gratefully thank to Allah SWT that blessing us every time and to the Department of Informatics, ITB that gives us a lot of inspiration to finish this research, not only its beautiful environment but also its friendly academicians. In addition, thanks to Dr. Rinaldi Munir and my family for the unexplainable helps.

REFERENCES

[1] Tingyuan Nie, Chuanwang Song and Xulong Zhi, "Performance Evaluation of DES and Blowfish Algorithms", IEEE International Conference on Biomedical Engineering and Computer Science (ICBECS- 2010), pp. 1-4, 23-25 Apr 2010.

[2] Prerna Mahajan, Abhishek Sachdeva. "A Study of Encryption Algorithms AES, DES and RSA for Security". Global Journal of Computer Science and Technology Network, Web & Security, ISSN: 0975-4350, Volume 13, Issue 15, Version 1.0, March 2013. Global Journals Inc. (USA)

[3] William Stallings, "Cryptography and etwork Security: Principles and Practice", Pearson Education/Prentice Hall, 5 th Edition.

[4] E. Thambiraja, G. Ramesh and Dr. R. Umarani, "A Survey on Various Most Common Encryption Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 7, pp. 226-233, July 2012.

[5] Diaa Salama Abdul. Elminaam, Hatem Mohamed Abdul Kader and Mohie Mohamed Hadhoud, "Performance Evaluation of Symmetric Encryption Algorithms", IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.12, pp. 280-286, December 2008.

[6] G. Carter, E. Dawsony and L. Nielseny, Key Schedule Classification of the AES Candidates, in Proceedings of the end AES Conference, Rome, Italy, 1999.

[7] Omari H. Ahmed, Al-Kasasbeh M. Basil, Al-Qutaish E. Rafa, Muhairat I. Muhammad, 2008, "A New Cryptographic Algorithm for the Real Time Applications ",Proceedings of the 7th WSEAS International Conference on INFORMATION SECURITY and PRIVACY (ISP '08)

[8] Stallings, W. (2000). "Network Security Essentials: Applications and Standards". Michigan:Prentice Hall

[9] C. E. Shannon "Communication Theory of Secrecy Systems", Bell Systems Tech. Jr. Vol 28, pages 656-715, 1949

[10] David Kahn, "THE CODEBREAKERS: The Story of Secret Writing," Macmillan publishing Co., Inc., 1967

[11] Feistel, Horst (1973). "Cryptography and Computer Privacy". Scientific American 228