Introductory Analysis of CALF Algorithm

A Dynamic-Function-Based Block Cipher Algorithm

Fauzan Hilmi Ramadhian
Informatics Engineering
Institut Teknologi Bandung
Bandung, Indonesia
fauzan_hilmi@hotmail.com

Calvin Sadewa
Informatics Engineering
Institut Teknologi Bandung
Bandung, Indonesia
master23680@gmail.com

***Abstract*—Encryption is an important and fascinating topic in computer security. It is used to convert data into an unrecognizable form that can only understood by authorized people. One of the best and mostly-used encryption technique is block cipher, a cipher in which the key and algorithm are applied to a block of data at once instead of one bit at a time. There are various types of block cipher exist nowadays. In this paper, CALF Algorithm, a new block cipher algorithm, will be discussed. There will be discussions about the algorithm's design, implementation, and it's security & histogram analysis.**

Keywords—encyrption, cipher, block cipher, CALF Algorithm

I. INTRODUCTION

Nowadays, computer security is an interesting topic to discuss. A fascinating branch of computer security is cryptography, the art of transforming message to a another message so that it is secure from unauthorized people. Cryptography is one of the cornerstone in securing modern communication. It is widely used in authentication, digital signature, and certification authority.

Within Cryptography, there lies cipher, a method or algorithm to convert plaintext into ciphertext. There are various types and kinds of cipher exist today. However, in general, there are two variants of cipher, stream based cipher and block based cipher. Stream based encrypt plain text by bit basis while block based encrypt plain text by block basis.

In block cipher, the size of a block generally is multiplicative of 8 bits. The exact algorithm of encrypting and decrypting is usually known and not dependent to the key. So, the block ciphers are generally strong in security. Some examples of block cipher are DES, IDEA, RC5 and AES. Although the algorithms have been proven secure, nothing is perfect in this world. The researchs to develop better block ciphers are still conducted until today. This paper is our contribution to the research of block cipher. There will be analitical discussion of CALF, a new block cipher which is developed by us. We hope this algorithm can contributes as much as possible to the world of cryptography.

II. THEORY

A. *Cipher*

Cipher is a method to transform a plaintext into another text using a key without losing the information contained in the plaintext. Usually, the cipher method is chosen as selective as possible so that the relation between the plaintext and the transformed text is blurred.

B. *Block Cipher*

Block cipher is a type of cipher which enciphers the message per block basis. Block cipher have various modes of operation, such as Electronic Code Book (ECB), Cipher Block Chaining (CBC), and Cipher Feedback (CFB) n-bit. ECB mode divided the message into blocks and encrypted the block individually. CBC first XOR the block with a block called initialization vector (IV), then encrypt it with key, the resulting encrypted block become IV for the next block. For the first block the IV is gained from user. In CFB n-bit, the IV is encrypted using the key, then the n-most significant bit from the result is XOR-ed with n-most significant bit from the plain block. The constraint is n must be less than the size of block in the original cipher.

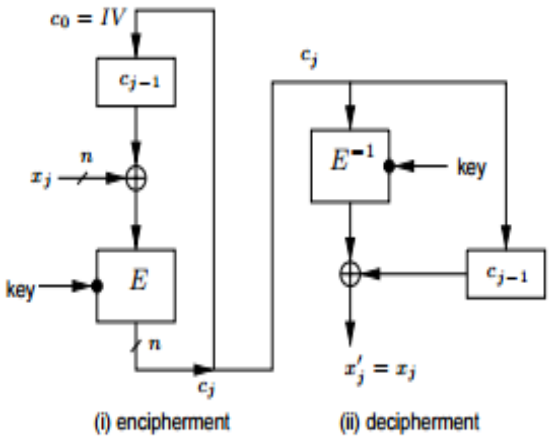


Fig 2.1. Illustration of how CBC works
[*Handbook of Applied Cryptography, by A. Menezes, P. van Oorschot, and S. Vanstone, CRC Press, 1996. P.229*]

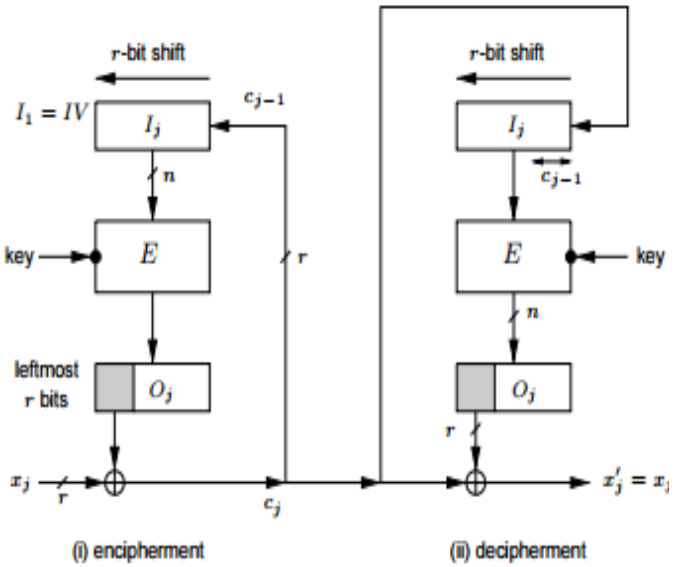


Fig 2.2. Illustration of how CFB works [*Handbook of Applied Cryptography, by A. Menezes, P. van Oorschot, and S. Vanstone, CRC Press, 1996. P.229*]

In ECB mode, a block is encrypted will always resulted same encrypted block. This makes a pattern that can be used to attack the cipher. In CBC mode, the pattern will be less occured than in ECB. This is because the IV. However, because the encrypting method need to use the previous result, it is not parallelizable. This drawback doesn't apply for decrypting method. CFB n-bit is almost the same with CBC, however the amount of bit encrypted in one time can be adjusted using n, making this kind of cipher flexible. CBC and CFB model also can be used to detect corruption in the data as in data may be tampered.

Application of block cipher is usually related to security. A message is encrypted so that an attacker who intercept the encrypted message doesn't know the actual information hidden without the actual key. However, due to varying platforms in which the cipher used, different needs arise so each cipher designed to meet the need. For example, in an environment where there is no re-programmable computer, one need to design an cipher that can be implemented using only circuit.

There are two common concepts in block cipher, confusion and diffusion. Confusion means that the resulting text should depends on the key, and diffusion means that a change in a bit must affected other bits in the ciphertext.

Another concept in cryptography is Feistel network. The basic idea of it is to split the encrypting method into several round. Then, split the plain block into Left side and Right side. In a round, the right side will be inserted into a round function with a round key. Next, the result will be XOR-ed with left side. This result is the new right side value while the previous right side will be the next left side. The round key itself is generated from the global key. Here is the depiction of a round of Feistel network.

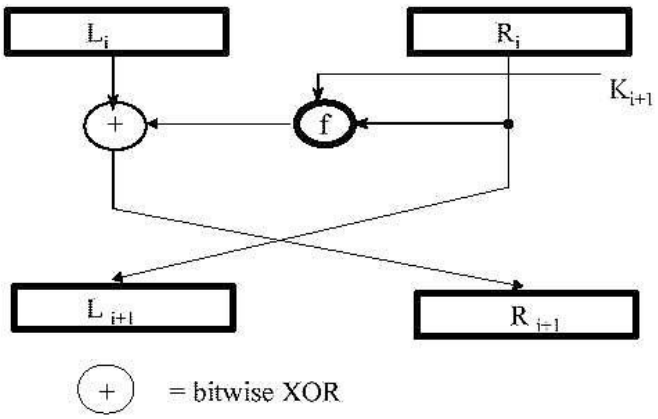


Fig 2.3. A diagram of Feistel network [*cryptome.org/jya/aes-rsa99.htm*, accessed on 17 March 2015, 05.47 pm]

One of the advantage of Feistel network is the decrypt stage is almost the same as encrypt stage. It makes the amount of code (or circuit in hardware implementation) can be suppressed. Another advantage is the round function need only to output result with same size as left side.

C. *MD5*

MD5 is a hash function that converts any-sized text into 128 bit hash value. MD5 is used in many cryptographic function as hash function. Although it has been proved that MD5 is not a secure hash function, it is one of the most widely used hash function until today.

D. *Cryptanalysis*

Cryptanalysis is a process of deciphering encrypted messages without knowing the key. Some people considered cryptanalysis as an art or game. For people who work on cryptography, cryptanalysis is considered as nemesis which must be defeated. As the field of cryptography evolves, the study of cryptanalysis is evolving too. Everyday, newer and better cryptanalysis method is developed. In general, there are 3 cryptanalysis type based on the cryptanalyst/attacker knowledge on the plaintext, key, and ciphertext . The types are:

1. Ciphertext-only attack
   This is a situation where the attacker only knows the ciphertext. He/she doesn't know anything about the plaintext or the key. This is the weakest situation of all three as the attacker barely knows anything. The only possible methods that can be used are brute-force attack and statistical attack. These are not good

methods since brute-force requires a long time and statistical attack is not always reliable all the time.

2. Known-plaintext attack
   In this situation, some pairs of ciphertext-key are known by the attacker. Although the information is still limited, this can be regarded as big improvement from the ciphertext-only attack. An example of reliable method to use in this situation is brute-forcing the key. Brute-forcing the key requires significantly less time than brute-forcing both the plaintext and the key in ciphertext-only condition.

3. Chosen-plaintext attack
   Chosen-plaintext attack is the most sophisticated condition of all three. Here, the attacker can freely chooses any plaintext and see the resulting ciphertext. His/her only job is to find the key. There are many cryptanalysis methods developed based on this conditon since chosen-plaintext attack is a lot easier than the other two.

III. PROPOSED ALGORITHM

A. Concept & Design

Upon designing CALF, we want to invent something new in block ciphers. We want to create more confusion and diffusion in the algorithm we are developing. Then, we got hit by an idea. Until now, every block ciphers we know is using fixed function in their Feistel network. Why not designing the function dynamic? So, a round in Feistel network can uses different function/operator than the other rounds. This is the fundamental idea of CALF design.

CALF is capable of encrypting arbitrary-sized plaintext with arbitrary-sized key too. Like other major block ciphers, CALF algorithm is Feistel-based. Here, Feistel round is done in 17 rounds. On each round, CALF is encrypting 128-bit of message block with 128-bit internal key to return a 128-bit ciphertext.

What makes CALF special is the *f* function. *F* function is a function/operator in Feistel network that takes in two inputs of right-half of message and the internal key. In traditional block cipher, *f* is fixed and known for all rounds. This is not the case in CALF algorithm. Here, *f* is unknown because it depends on the input. The complete execution of *f* is firstly, the right-half part of message is processed with sub-bytes-process. Then, the result is combined with the already-64-bit-converted internal key by the *Op* function/operator. *Op* is the part of *f* which is customizable and undeterminable. There are 4 possibilities of *Op.* They are :

1. XOR
2. Addition then 2 times of MOD 2^{32}
3. XOR then addition then 2 times of MOD 2^{32}
4. Addition then 2 times of MOD 2^{32} then XOR

In practice, there may be more than these *Op* possibilities. However, for now, CALF is only using those 4 operators. After that, a question arises; Which operator is used? To determine *Op*, the right-half bytes is MOD-ed by 4 to get the operator number. Again, the ordering of operators is customizable. Beside its use in *f, Op* is also used in key shrinking process. The detail of this process will be explained later. Back to *f*, after *Op* is done, the final step is executed. The final operation of *f* consists of XOR-ing the result of *Op* with the left-half of message.

B. Encipher Technique

Basically, the encipher process consists of block partition and block processing. Block partition is the process of partitioning the plaintext into 128-bit individual blocks. It is a possibility that the last block is less than 128-bit. To handle this, a number of zeroes are padded in the back of the block. Then, each block is processed with the Feistel rounds. In *ECB* (*Electronic Code Book*) or normal mode, each block is running exclusively. However, in *CBC* (*Cipher Block Chaining*) mode, a block is depending on its previous block. For i={1,2,...,n-1}, before running the block operation, block(i) is XOR-ed with the already-processed block(i-1). In *CFB* (*Cipher Feedback Block*) mode, a block is depending on other blocks too, with only slight difference compared to CBC.

Now, the block process is started. The steps are:

1. Internal key generation
   The internal key is generated with MD5 function. The key of each round is the result of MD5 of it's previous round' key. But, there is exception for round 1. Round 1 uses the global (inputted) key as the input.

2. Block partition
   Now, the Feistel round is started. The input block is divided into two halves, each sized 64-bit. From now, the left part and right part of block will be referred as *L* and *R*, respectively.

3. Determine *Op*
   Op is determined based on the operation *R* mod 4. The result is the number of operation chosen. The list of operations are:
   i. *XOR*
   ii. Addition then 2 times of MOD 2^{32}
   iii. XOR then addition then 2 times of MOD 2^{32}
   iv. Addition then 2 times of MOD 2^{32} then XOR

4. Internal key shrinking
   This process is used to convert the 128-bit key into 64-bit. How? First, the key is splitted into two halves. Then, both halves are *Op*'ed to get the shrinked key. From now, this shrinked key will be referred simply as *key*.

5. SubBytes
   Here, *R* will be transformed using an S-Box. CALF is using the same S-Box as Rijndael's which is used in AES. The technique is similar too. Each byte (two hex block) of *R* is converted by lookup-table of the S-Box. The S-Box itself is pictured below.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Fig 3.1. Rijndael's S-Box. [*samiam.org/s-box.html*, accessed on 17 March 2015, 03.12 pm]

6. *Op R* and *key*
   The resulting *R* from previous step now is *Op'ed* with the *key*.

7. XOR and position switching
   In this process, *L* and *R* is XOR'ed. The result becomes new *R (R')* and the old & unchanged *R* becomes new L (L').

Later, the Feistel round is conducted again with the input block is *L'* and *R'*. This is repeated for 16 more times. Then, the final *L'* and *R'* after round 17 is combined. It is the result of an encipher process of a block with CALF algorithm.

For a clearer picture, here is the Feistel network diagram of CALF.

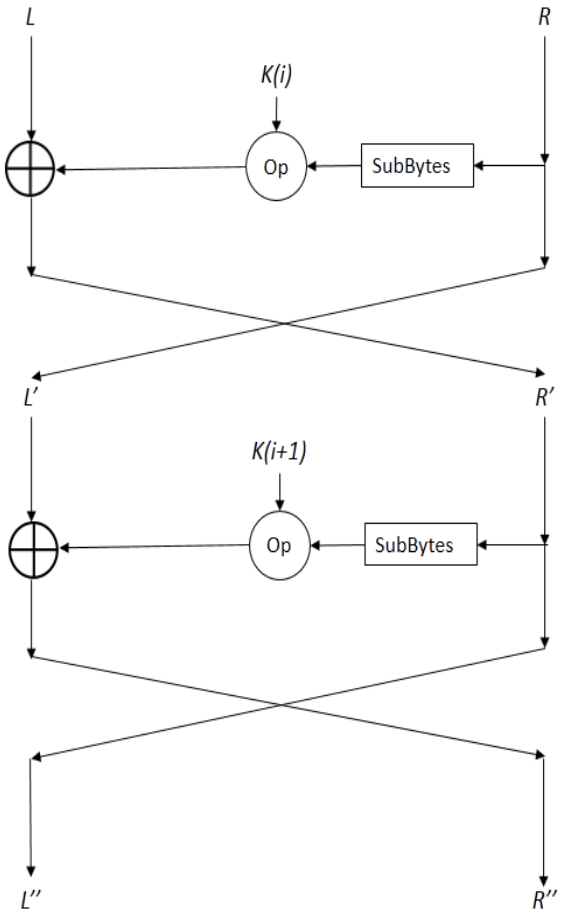


Fig 3.2. CALF's Feistel Network Diagram

C. Decipher Technique

The ciphertext and global key are required to successfully completed the deciphering process. Basically, deciphering CALF is reversing the Feistel rounds. It is really similar to the encipher procedure with the only notable difference is the reversal of *L* and *R*. So, the decipher process consists of steps 1 to 7 of encipher's, but *Op* is determined by *L* mod 4. Then, *L* is transformed with SubBytes and *Op'*ed with *key*. After that, *R* is XOR'ed with *L*. The result becomes new L (L') and the old & unchanged *R* becomes new R (R'). Later, these procedures are executed 16 more times. Then, the final *L'* and *R'* combined into the resulting plaintext block.

IV. ANALYSIS

A. Security Analysis

The security analysis is conducted based on 3 possible cryptanalysis conditions that already discussed in Theory section. Here are them.

1. Ciphertext-only attack

   The attacker is in really hard situation. Ciphertext-only attack is hard, and it is a lot harder for block cipher. Statistical attack is almost impossible since the bytes in block cipher are distributed uniformly. The brute-force attack is not good either. Assume that a hacker is trying to break a block (128-bit) of CALF-ciphered ciphertext. He/she must try 2^{128} possibilities of 128-bit MD5-resulted key. Assuming ~2 million keys can be tested in a second, there is still needed about 26 quadrillion years to compute all possible keys. And it is done without knowing what is the right plaintext. So, choosing the semantically-possible-plaintext takes quite long time too. Then, if the original key is needed, the hacker still needs to break MD5 function. So, again, another not-a-short time is needed.

   After observing the ridiculously long time to break only a block of message, it can be safely said that CALF can stand the ciphertext-only attack.

2. Known-plaintext attack

   The brute-force attack still used on this condition. However, it is easier than the one in ciphertext-only attack. Now, the brute force is focused solely on determining the key as the plaintext is already known. It can be more easier if multiple pairs of ciphertext-plaintext are known. However, there is still needed a lot of time to do that. Assuming 1000 pairs are known, 25 trillion years still needed to generates all possible keys. Although it is better than ciphertext-only attack, this attack still requires infeasible time. Thus, it can be said that CALF is safe from known-plaintext attack.

3. Chosen-plaintext attack

   Now, the attacker is in significantly better condition than the previous two. He/she can uses various combinations of plaintext and key to get desired information. Here will be showed some examples of input and output of the CALF cipher, especially the one that is "dangerous" to exploited in chosen-plaintext attack

a. Ordinary plaintext and key

> Input : Lorem ipsum dolor sit amet, nam rebum fugit alterum eu. Id vidit delenit urbanitas vis, regione electram pri ex, dicam consul causae sit te.
> Key : kriptografi rata
>
> Ciphertext : □ ꭑ홰 ꞵ ȧᾳ̃↘筳
> □̏E• ጵ縩⁝ ᛉ欲ℋ婧൭□៖6□堯褻鈆▬⚿撞屎
> □栐㳂杞虢腴写䪻䍽琎租
> 筊䗇⁝□倉蟈□器䍢□殓참琗□ৢ⇆鐇恆□闟铰
> ㄴ隱軲곢䜌潚着㖨䁖쐴
>
> Decipher result :
> Lorem ipsum dolor sit amet, nam rebum fugit alterum eu. Id vidit delenit urbanitas vis, regione electram pri ex, dicam consul causae sit te.

For an ordinary input and key, CALF returns completely different ciphertext. The ciphertext is unrecognizable when compared to the plaintext and the key. Also, it is successfully decrypted back. Thus, it can be said that CALF converts the plaintext into an unrecognizable ciphertext, but it can sucessfully decipher back the ciphertext.

b. Slightly different plaintext than case (a)

> Input : Morem ipsum dolor sit amet, nam rebum fugit alterum eu. Id vidit delenit urbanitas vis, regione electram pri ex, dicam consul causae sit te.
> Key : kriptografi rata
>
> Ciphertext : □ ꭑ홰 ꞵ ʃᾳ̃↘筳
> □̏E• ጵ縩⁝ ᛉ欲ℋ婧൭□៖6□堯褻鈆▬⚿撞屎
> □栐㳂杞虢腴写䪻䍽琎租
> 筊䗇⁝□倉蟈□器䍢□殓참琗□ৢ⇆鐇恆□闟铰
> ㄴ隱軲곢䜌潚着㖨䁖쐴

Now, the plaintext is slightly changed compared to (a) as the first letter is 'M' not "L'. The result of this little modification is almost unrecognizable. Ciphertext in (a) almost identical ot (b). This is not good as the attacker can exploited this property to predicting the key by chosen-plaintext attack. However, it still takes a big effort to determining the key as the difference between ciphertexts is hardly predictable.

c. Slightly different key than case (a)

Input : Morem ipsum dolor sit amet, nam rebum fugit alterum eu. Id vidit delenit urbanitas vis, regione electram pri ex, dicam consul causae sit te.
Key : lriptografi rata

Ciphertext : □욨□□塗畾□碻砮鞞두ⓤゴ榧趰 峌翍쬣좼鮲ؤ㉆九旪╠※□襟굅詠⊝櫋鼿抂 □잇鑿쳟쉾쨌姙蒝梾ם□螬뮏轊ב翧愣 뾇누횔씠ݣ ㅁㅅ 쐥鸕⚲◌ා顠聊⸫諺鬻□篷ᵕ湇窔

In this case, the key is slighly different from (a) as its first letter is ‘l’ not ‘k’. Even though the change is small, it makes the ciphertext really different than (a). The reason why this happens is because different key gives different internal keys (MD5 result), even though the difference is hardly recognizable. So, a little change in key makes big difference in ciphertext. This fact makes the attacker’s job a lot harder.

d. Blank plaintext

Sometimes, blank ciphertext is a neat trick that can be used by the attackers. Some cipher algorithms have property that when blank plaintext is inputted, the ciphertext is *equal* to the key (i.e. Vigenere Cipher). This is really dangerous property that can be easily exploited. Does CALF has this hazardous trait?

Input :
Key : kriptografi rata

Ciphertext :

Turns out that when the plaintext is blank, regardless of the key, the ciphertext outputted is always blank too. It means that CALF doesn’t have the dangerous “blank plaintext” trait .

e. Repeated block

This property is a big weakness of block ciphers and thus is exploited by the attackers. Basically, in ECB mode, if a block of message is converted into block cipher *BC*, then other same blocks of message will be surely ciphered into *BC* too. Does this happen in CALF too?

Input : 123456789012345X
Key : kriptografi rata

Ciphertext :

Input : 123456789012345X123456789012345X
Key : kriptografi rata

Ciphertext :
Ö«=“$Ñõ£ÐRvn8ÀºÖ«=“$Ñõ£ÐRvn8Àº

An interesting phenomenon can be seen above. In the first picture, a 128-bit (a block) plaintext is inputted so that the resulting ciphertext is outputted. If the block is repeated, the ciphertext is repeated for each block too. But, the text pattern is different. In fact, if the plaintext size is increased again, the pattern will be different again although each block is always repeated. This is an unique property of CALF as other block ciphers rarely change pattern when the input size is getting bigger. This trait has positive effect too, as attacker is facing hard obstacle to utilizie repeated block trick to break CALF. In conclusion, CALF has a special property that makes it harder to break with repeated block method.

A better result can be obtained if the encipher mode is not ECB. In ECB, each block is enciphered independent of each other. Thus, repeating blocks can easily gives clue about the inputted key. To handle that, previous-block-dependent-modes CBC and FCB are used. Here are the results.
With CBC :

Input : 123456789012345X
Key : kriptografi rata

Ciphertext : ê;ž]ªMÓÌö¡H°

Input : 123456789012345X123456789012345X
Key : kriptografi rata

Ciphertext : ñ Ýõ±#QÎÐ‹ÿ×Ò$“Ê¬|{Tpë

With CFB:

Input : 123456789012345X
Key : kriptografi rata

Ciphertext : ¹Ðî<%=ÓóñÿÝæP®

Input :123456789012345X123456789012345X
Key : kriptografi rata

Ciphertext : î$Ä+•tq³¾ã³d@`¾ëœsHÎg4º…yå;

It can be seen that CBC and FCB give more randomized characters distribution on their ciphertext. More importantly, the blocks in the ciphertext are not repeated or showing any

sign of patterns even though the plaintext's blocks are clearly repeated. Thus, CBC and CFB is better method to use when enciphering, particularly when the message blocks are repeated. Specifically, when CALF is used in data communication, CFB is preferred because its ability to enciphering data in less than one-block size.

B. Histogram Analysis

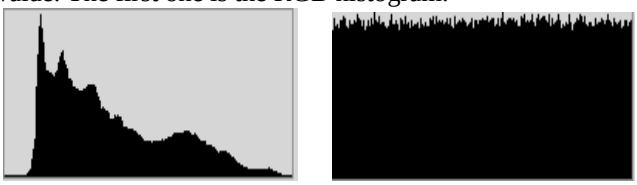
Histogram is a graph that can show distribution of data. A special type of histogram is image histogram. In image histogram, the data is the pixel values in the image. Histogram can be used as an indicator in frequency analysis, which in turn can be used to guess what the key is. To prevent this, the encryption algorithm should blur the statistical relation between the source image and the encrypted image. One way to achieve this is to make the encrypted image histogram distribution show same distribution for any data.

Below is image before and after encryption

a. Before Encryption *b. After Encrytion*

Fig. 4.1. Original picture of Lena (left) and after it is enciphered with CALF (right)

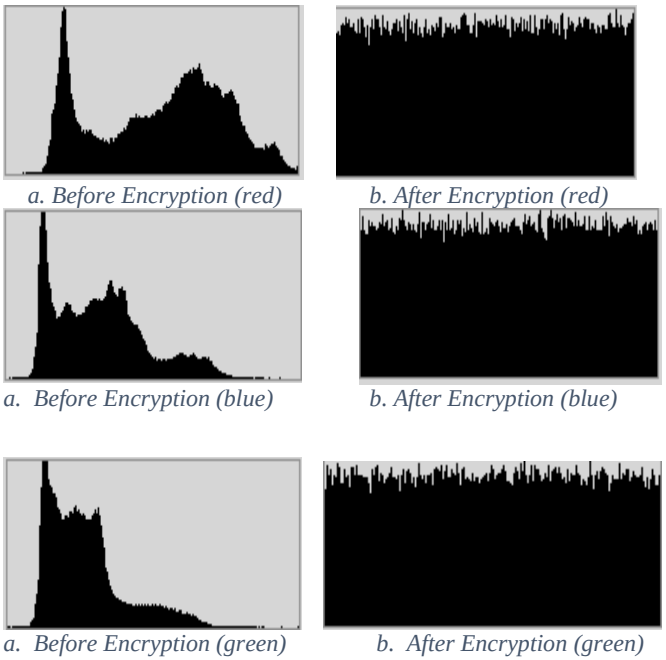
Now, the histogram of above images will be analyzed. These histograms are extracted using Adobe Photoshop CS5. Note that the value of histogram is relative to the maximal value. The first one is the RGB histogram.



a. Before Encryption (RGB) *b. After Encryption (RGB)*

Fig. 4.2. RGB histogram of original Lena (left) and after encryption with CALF (right)

Here are the histograms for the each dimension (red, green, blue) of RGB.



a. Before Encryption (red) *b. After Encryption (red)*

a. Before Encryption (blue) *b. After Encryption (blue)*

a. Before Encryption (green) *b. After Encryption (green)*

Fig. 4.2. Red, Green, and Blue histograms of original Lena (left) and after encryption with CALF (right)

From all of the histograms, a simple observation can be conducted. Before encryption, the RGB distribution is not uniform. However, after CALF takes place, the encrypted image is flat and distributed on each colour dimension and in RGB total. Thus, it is hard to do the statistical attack or frequency analysis to the cipher image as the colour divergence is so small. So, CALF is strong in defence against statistical attack to cipher image.

V. CONCLUSIONS & FUTURE WORKS

Cryptography is an important topic in computer security. One type of most widely used cryptography technique is block cipher. There are various kinds and types and block cipher. In this paper, CALF algorithm, a new block cipher, is proposed.

CALF's basic concept is to create more confusion and diffusion than other existing block ciphers. It uses customizable and unguessable part called *Op*. Besides that, CALF is not much different than other Feistel-based block ciphers.

In experiment, it is proven that CALF is quite strong in security analysis. The ciphertext outputted has good randomness and uniformity among the characters. It is also resistant to various attack conditions, expecially blank plaintext and repeated block. It is highly suggested to use CBC or CFB mode instead of ECB to handle the repeated block case better. However, this algorithm still quite vulnerable to slightly-different plaintext method in chosen-plaintext attack.

CALF got good result in histogram analysis too. The enciphered image has good uniformity among the colours. Thus, CALF is pretty resistant to statistical attack or frequency analysis.

In future, a deeper and more comprehensive research can be conducted. CALF has room for customizations and improvements. One can analyze what is the impact of changing algorithm's number of rounds, number of operators, and type of operators. Or, new block cipher algorithm may be developed based on CALF algorithm. We hope that this algorithm can contribute as much as possible to the study of cryptography in general, and to the study of block cipher in particular.

ACKNOWLEDGMENT

First of all, Authors would say thank you to Almighty God because of His mercy and grace Authors can finish this paper. Then, Authors also wants to express their thanks to Dr. Ir. Rinaldi Munir, M.T. whose give helpful advices and assistances. Finally, Authors want to say thank you to their parents and beloved friends who are always give Authors strengths and spirits to pass the struggles during the writing of this paper.

REFERENCES

[1] Fayoumi, Hiba. *Cryptanalysis.* http://www.math.colorado.edu/~hiba/crypto/cryptanalysis.html, accessed on 17 March 2015, 07.35 pm

[2] Munir, Rinaldi. 2012. *Security Analysis of Selective Image Encryption Algorithm Based on Chaos and CBC-like Mode.* 2012 7th International Conference on Telecommunication Systems, Services, and Applications (TSSA)

[3] Rouse, Margaret. *Block Cipher.* http://www.searchsecurity.techtarget.com/definition/block-cipher, accessed on 17 March 2015, 07.10 pm

[4] Trenholme, Sam. *S-box Used by the AES Cryptographic Algorithm.* http://www.samiam.org/s-box.html, Accessed on 17 March 2015, 06.34 pm