Geldy : A New Modification of Block Cipher

Candy Olivia Mawalim (13513031)
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung 40132, Indonesia
13513031@std.stei.itb.ac.id

Angela Lynn (13513032)
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung 40132, Indonesia
13513032@std.stei.itb.ac.id

Abstract—**One of the important issues in computer security is encryption algorithm. This is a process of encoding message or information, which is called plain text to a message that cannot be easily read by other people without the key (cipher text). There are many variation of encryption algorithm. Block cipher is one of these algorithms which is quite popular to be developed. In this paper, we propose a variation of block cipher algorithm named Geldy Algorithm. This algorithm used substitution and Feistel Network method. The plain text is transformed to and proceeded in bit.**

Keywords—encryption algorithm; block cipher; Geldy Algorithm, substitution; Feistel Network; bit.

I. Introduction

Nowadays, security becomes an interesting issue to discuss. Security concerns to privacy, authenticity and identity of a person or an organization. To ensure the security of a channel we need cryptography. Cryptography is the discipline that use cyphers to protect secrets. It has been developed since the Old Kingdom of Egypt in 1900 BC. During the Middle Ages, Cryptography became more popular. There were many inventions of new encryption and decryption algorithm, for example Vigenère Ciphers, Uesugi Cipher and Playfair Cipher. After that era, with the advancement of communication technology, encryption and decryption algorithm developed rapidly. Encrypting and decrypting ciphers in Internet Era has shifted from machines to computers.

The use of computers in modern era caused the way of encrypting and decrypting based on bit representation. Generally, the procedures started with transforming message to blocks of bits. These blocks of bits then modify with bit based cipher algorithm. There are two category of bit based cipher algorithm, such as stream cipher and block cipher. Stream cipher operates bit per bit messages however block cipher operates block of bits.

The size of block in block cipher algorithm generally is an 8 bits multiplication block. The operation mode of block cipher can be divided into 4 modes : Electronic Code Block (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), and Output Feedback (OFB). Each mode has its own positive and negative side. Beside of the operation modes, design principles of block cipher can really influence the secure level of this cipher algorithm. There are four principles in designing block cipher : (1) Confusion and Diffusion from Shannon, (2) Iterated Cipher, (3) Feistel Network, and (4) S-Box. In spite of block cipher algorithms is quite secure, researchers still develop better block cipher algorithms. This paper is our contribution to the research of block cipher. There will be analytical discussion of Geldy, a new modification of block cipher. We hope this algorithm can contributes to the development of cryptography.

II. Theory

A. *Modern Cryptography*

Modern cryptography is a cryptography algorithm which operated in bit mode. Key, plain text, as well as cipher text, all are processed in bit. Since it is in bit mode, XOR operation is the most common used to encrypt or decrypt. Modern cryptography is supported by the using of digital computer for message confidelity, because data is represented in binary. Like classic cryptography, modern cryptography also use substitution and transposition as concept to encrypt or decrypt, but in a harder way. The substitution and transposition is more complex.

Modern cryptography is the cornerstone of computer and communications security. Its foundation is based on various concepts of mathematics such as number theory, computational-complexity theory, and probability theory. In modern cryptography, usually message is converted to bit form and then divided to some blocks. For example, 100111010110 is divided to 4-bit blocks, become 1001, 1101, and 0110. Then, those blocks are represented in hexadecimal to make it more complicated and hard to solved.

There are three major characteristics that separate modern cryptography from classic cryptography.

1. It operates on binary bit sequences.
2. It relies on publicly known mathematical algorithms for coding the information.
3. It requires parties interested in secure communication to possess the secret key only.

There are two types of bit-based algorithm.

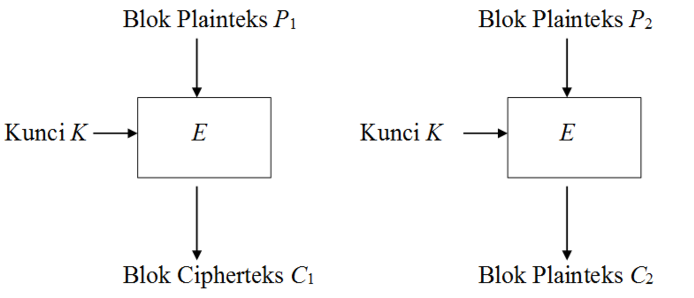
1. Stream Cipher, an algorithm which operated on single bit.
2. Block Cipher, an algorithm which operated on block of bit.

This paper will discuss more about Block Cipher algorithm.

B. Block Cipher

A block cipher is a function which maps n-bit plain text blocks to n-bit cipher text blocks (n = block length). It may be viewed as a simple substitution cipher with large character size. The function is parameterized by a k-bit key K, taking values from a subset k (the key space) of the set of all k-bit vectors Vk. It is generally assumed that the key is chosen at random. Plain text and cipher text has equal size to avoid data expansion.

To allow unique decryption, the encryption function must be one-to-one (i.e., invertible). For n-bit plain text and cipher text blocks and a fixed key, the encryption function is a bijection, defining a permutation on n-bit vectors. Each key potentially defines a different bijection. This process can be illustrated and formulated as Figure 1.



(a)

$$E : V_n \times k \rightarrow V_n$$

Such that for each key $K \in k, E(P,K)$ is an invertible mapping from V_n to V_n, written Ek (P).

(b)

Figure 1.(a) Illustration of Block Cipher (b) Formulation of Block Cipher

There are four most common modes of operation in encrypting and decrypting block cipher (ECB, CBC, CFB, and OFB).

i. ECB (Electronic codebook) mode
The pseudo code of this mode is shown below.

```
INPUT   : k-bit key K; n-bit plain
text blocks x1,x2,...,xt.
SUMMARY : produce cipher text blocks
c1,c2,...,ct; decrypt to recover
plain text.
   (1)     Encryption :
      $1 \le j \le t,\ c_j \leftarrow E_K(x_j).$

   (2)     Decryption :
      $1 \le j \le t,\ x_j \leftarrow E_K^{-1}(c_j).$
```

Properties of the ECB mode of operation :

1. Identical plain text blocks (under the same key) result in identical cipher text.
2. Chaining dependencies: blocks are enciphered independently of other blocks. Reordering cipher text blocks results in correspondingly re-ordered plain text blocks.
3. Error propagation: one or more bit errors in a single cipher text block affect decipherment of that block only. For typical ciphers E, decryption of such a block is then random.

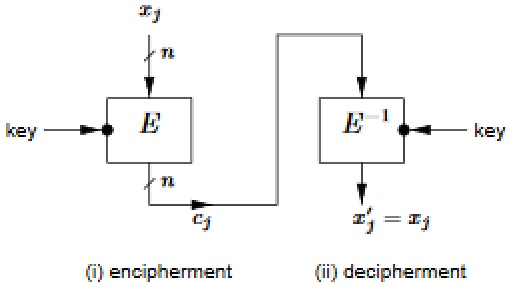


Figure 2. Electronic Codebook (ECB)

source : *Handbook of Applied Cryptography by A. Menezes, P. van Oorschot and S. Vanstone.*

ii. CBC (Cipher-Block Chaining) mode
The pseudo code of this mode is shown below.

```
INPUT   : k-bit key K; n-bit plain
text blocks x1,x2,...,xt.
SUMMARY : produce cipher text blocks
c1,c2,...,ct; decrypt to recover
plain text.
    (1)Encryption :
    $c_0 \leftarrow IV$. For $1 \le j \le t,\ c_j \leftarrow E_K(c_{j-1} \oplus x_j).$

    (2)Decryption :
    $c_0 \leftarrow IV$. For $1 \le j \le t,\ x_j \leftarrow c_{j-1} \oplus E_K^{-1}(c_j).$
```

Properties of the CBC mode of operation :

1. Identical plaintexts: identical ciphertext blocks result when the same plaintext is enciphered under the same key and IV. Changing the IV, key, or first plaintext block (e.g., using a counter or random field) results in different ciphertext.
2. Chaining dependencies: the chaining mechanism causes ciphertext cj to depend on xj and all preceding plaintext blocks (the entire dependency on preceding blocks is, however, contained in the value of the previous ciphertext block). Consequently, rearranging the order of ciphertext blocks affects decryption. Proper decryption of a correct ciphertext block requires a correct preceding ciphertext block.
3. Error propagation: a single bit error in ciphertext block cj affects decipherment of blocks cj and cj+1 (since xj depends on cj and cj−1). Block x0j recovered from cj is typically totally random (50% in error), while the recovered plaintext x0j+1 has bit

errors precisely where cj did. Thus an adversary may cause predictable bit changes in xj+1 by altering corresponding bits of cj.

4. Error recovery: the CBC mode is self-synchronizing or ciphertext autokey in the sense that if an error (including loss of one or more entire blocks) occurs in block cj but not cj+1, cj+2 is correctly decrypted to xj+2.

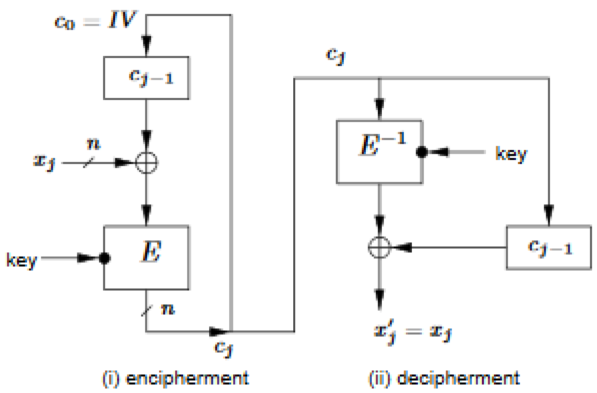


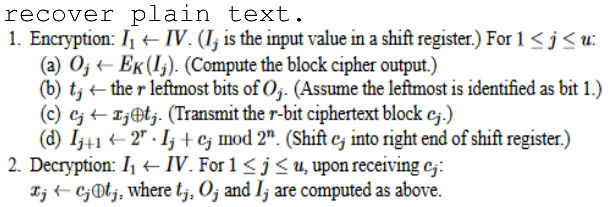
Figure 3. Cipher-block Chaining

source : *Handbook of Applied Cryptography by A. Menezes, P. van Oorschot and S. Vanstone.*

iii. CFB (Cipher Feedback) Mode

The pseudo code of this mode is shown below.

```
INPUT   : k-bit key K; r-bit plain
text blocks x1,x2,...,xu. $(1 \le r \le n)$.
SUMMARY : produce r-bit cipher text
blocks c1,c2,...,cu; decrypt to
recover plain text.
```



```
1. Encryption: $I_1 \leftarrow IV$. ($I_j$ is the input value in a shift register.) For $1 \le j \le u$:
    (a) $O_j \leftarrow E_K(I_j)$. (Compute the block cipher output.)
    (b) $t_j \leftarrow$ the $r$ leftmost bits of $O_j$. (Assume the leftmost is identified as bit 1.)
    (c) $c_j \leftarrow x_j \oplus t_j$. (Transmit the $r$-bit ciphertext block $c_j$.)
    (d) $I_{j+1} \leftarrow 2^r \cdot I_j + c_j \bmod 2^n$. (Shift $c_j$ into right end of shift register.)
2. Decryption: $I_1 \leftarrow IV$. For $1 \le j \le u$, upon receiving $c_j$:
    $x_j \leftarrow c_j \oplus t_j$, where $t_j$, $O_j$ and $I_j$ are computed as above.
```

Properties of the CFB mode of operation :

1. Identical plain texts: as per CBC encryption, changing the IV results in the same plaintext input being enciphered to a different output. The IV need not be secret (although an unpredictable IV may be desired in some applications).
2. Chaining dependencies: similar to CBC encryption, the chaining mechanism causes cipher text block cj to depend on both xj and preceding plain text blocks; consequently, reordering cipher text blocks affects decryption. Proper decryption of a correct cipher text block requires the preceding [n/r] cipher text blocks to be correct (so that the shift register contains the proper value).
3. Error propagation: one or more bit errors in any single r-bit ciphertext block cj affects the decipherment of that and the next [n/r] cipher text blocks (i.e., until n bits of ciphertext are processed, after which the error block cj has shifted entirely out of the shift register). The recovered plaintext x0j will differ from xj precisely in the bit positions cj was in error; the other incorrectly recovered plaintext blocks will typically be random vectors, i.e., have 50% of bits in error. Thus an adversary may cause predictable bit changes in xj by altering corresponding bits of cj.
4. Error recovery: the CFB mode is self-synchronizing similar to CBC, but requires [n/r] ciphertext blocks to recover.
5. Throughput: for r<n, throughput is decreased by a factor of n/r (vs. CBC) in that each execution of E yields only r bits of ciphertext output.

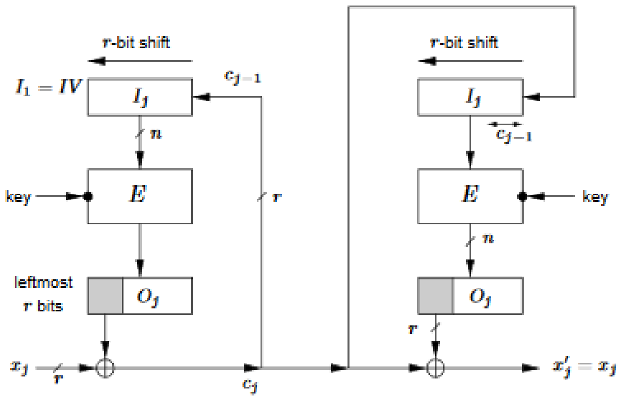


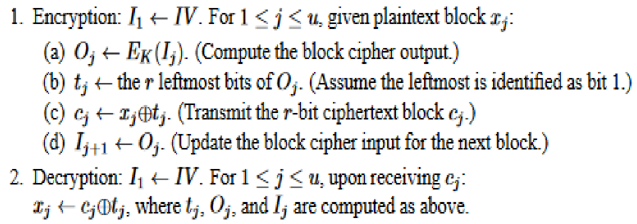
Figure 4. CFB, r-bit characters/ r-bit feedback

source : *Handbook of Applied Cryptography by A. Menezes, P. van Oorschot and S. Vanstone.*

iv. OFB (Output Feedback) Mode

The pseudo code of this mode is shown below.

```
INPUT   : k-bit key K; n-bit IV, r-bit
plain text blocks x1,x2,...,xu.
$(1 \le r \le n)$.
SUMMARY : produce r-bit cipher text
blocks c1,c2,...,cu; decrypt to
recover plain text.
```



```
1. Encryption: $I_1 \leftarrow IV$. For $1 \le j \le u$, given plaintext block $x_j$:
    (a) $O_j \leftarrow E_K(I_j)$. (Compute the block cipher output.)
    (b) $t_j \leftarrow$ the $r$ leftmost bits of $O_j$. (Assume the leftmost is identified as bit 1.)
    (c) $c_j \leftarrow x_j \oplus t_j$. (Transmit the $r$-bit ciphertext block $c_j$.)
    (d) $I_{j+1} \leftarrow O_j$. (Update the block cipher input for the next block.)
2. Decryption: $I_1 \leftarrow IV$. For $1 \le j \le u$, upon receiving $c_j$:
    $x_j \leftarrow c_j \oplus t_j$, where $t_j$, $O_j$, and $I_j$ are computed as above.
```

Properties of the OFB mode of operation :

1. Identical plaintexts: as per CBC and CFB modes, changing the IV results in the same plaintext being enciphered to a different output.
2. Chaining dependencies: the keystream is plaintext-independent.

3. Error propagation: one or more bit errors in any ciphertext character cj affects the decipherment of only that character, in the precise bit position(s) cj is in error, causing the corresponding recovered plaintext bit(s) to be complemented.
4. Error recovery: the OFB mode recovers from ciphertext bit errors, but cannot self-synchronize after loss of ciphertext bits, which destroys alignment of the decrypting keystream (in which case explicit re-synchronization is required).
5. Throughput: for r<n, throughput is decreased as per the CFB mode. However, in all cases, since the keystream is independent of plaintext or ciphertext, it may be pre-computed (given the key and IV).

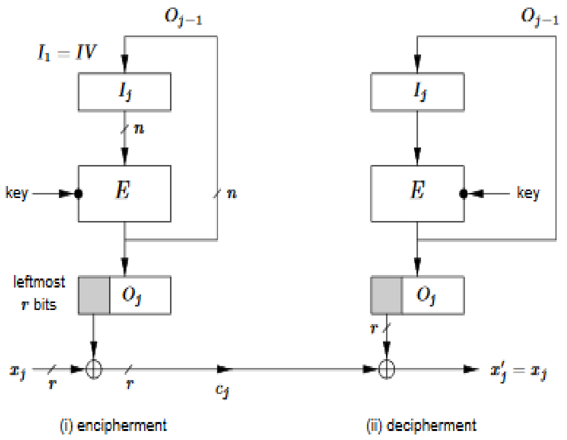


Figure 5. OFB, r-bit characters/ n-bit feedback

source : *Handbook of Applied Cryptography by A. Menezes, P. van Oorschot and S. Vanstone.*

Beside mode of ciphertext, we also need to know the principles of designing a block cipher. There are four principles in designing block cipher : (1) Confusion and Diffusion from Shannon, (2) Iterated Cipher, (3) Feistel Network, and (4) S-Box. In this paper, we will combine diffusion and confusion principle from Shannon and Feistel Network.

C. Substitution

Substitution is an encryption method by which units of plaintext are replaced with ciphertext based on a fixed system (subtitution or permutation table). Substitution ciphers can be compared with transposition ciphers. In this cipher, the units of plaintexts are rearranged in a different and complex order. This order is generated by a key with some iterated steps.

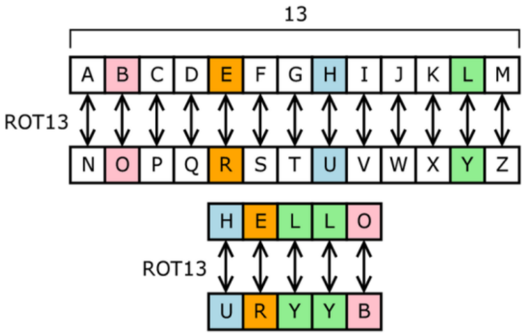


Figure 6. Simple Substitution Cipher

source : *https://en.wikipedia.org/wiki/Substitution_cipher#/media/File:ROT13.png*

D. Feistel Network

Feistel network is a symmetric structure used in the construction of block ciphers. Feistel structure has the advantage that encryption and decryption operations are very similar, requiring only a reversal of the key schedule. Feistel construction is iterative in nature which makes implementing in hardware easier. Feistel networks combine multiple rounds of repeated operations, such as bit-shuffling, simple non-linear functions, and linear mixing using XOR.

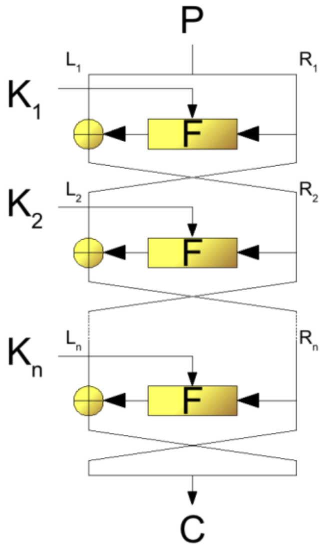


Figure 7. Feistel Diagram

III. Proposed Algorithm

The proposed algorithm consists of two phases, the encipher and decipher algorithm. These two sub-algorithms will be explained separately.

A. Encipher Technique

Generally, the encipher process consists of block partition and block processing. Block partition is the phase when the plaintext is divided into some number of blocks (with the block size = key block size). This blocks contain bits that represent the plaintext.

Block processing phase there are two steps i.e. substitution step and Feistel diagram step. We use three different substitution before continue to Feistel diagram step.

1. First of all, we swipe plaintext blocks. The odd elements of these block is swiped to left otherwise the even elements of these block is swiped to right. Odd or even elements is decided by the position of the elements (the blocks is arranged sequentially).

```
Result1 : convert_to_bitBlocks ⟶
shift_odd_to_left ⟶ shift_even_to_right
```

e.g. :

plaintext = “BUNGAROS”
Plaintext bit block :

```
0 1 0 0 0 0 1 0
0 1 0 1 0 1 0 1
0 1 0 0 1 1 1 0
0 1 0 0 0 1 1 1
0 1 0 0 0 0 0 1
0 1 0 1 0 0 1 0
0 1 0 0 1 1 1 1
0 1 0 1 0 0 1 1
```

Figure 9. Plaintext block

Odd shifted block :

```
0 1 0 0 1 0 0 0
0 1 0 1 0 1 0 1
0 1 1 0 1 1 0 0
0 1 0 0 1 1 0 1
0 1 0 0 0 0 0 1
0 1 0 1 1 0 0 0
0 1 1 0 1 1 0 1
0 1 0 1 1 0 0 1
```

Figure 10. Odd shifted block

Even shifted block:

```
0 1 0 1 1 0 0 0
0 0 0 1 0 1 0 1
0 1 1 1 1 0 0 1
0 0 0 1 1 0 0 1
0 1 0 1 0 0 0 0
0 1 0 1 1 1 0 0
0 0 1 1 1 0 0 1
0 1 0 1 1 1 0 0
```

Figure 11. Even shifted block

2. Next, shifted/ swiped plaintext blocks is operated with XOR operation. We generate prime blocks with the same block size. Prime blocks are blocks that consist of default element (zero) if not prime and one is prime.

```
Result2 = PrimeKey_n XOR ShiftedBlock_n
```

PrimeKey 1	PrimeKey 2

0 1 1 0 1 0 1 0
0 0 1 0 1 0 0 0
1 0 1 0 0 0 1 0
0 0 0 0 1 0 1 0
0 0 0 0 1 0 0 0
1 0 1 0 0 0 1 0
0 0 0 0 1 0 0 0
0 0 1 0 1 0 0 0
	0 0 1 0 0 0 1 0
	1 0 0 0 0 0 1 0
	0 0 1 0 0 0 0 0
	1 0 0 0 0 0 0 0
	1 0 0 0 1 0 1 0
	0 0 1 0 1 0 0 0
	1 0 0 0 0 0 0 0
	0 0 0 0 0 0 1 0

PrimeKey 3	**PrimeKey 4**

0 0 1 0 0 0 0 0
1 0 1 0 0 0 0 0
0 0 0 0 1 0 1 0
0 0 0 0 1 0 0 0
0 0 1 0 0 0 1 0
0 0 0 0 1 0 0 0
0 0 1 0 1 0 0 0
0 0 0 0 0 0 1 0
	1 0 0 0 1 0 1 0
	0 0 0 0 0 0 0 0
	0 0 1 0 0 0 0 0
	0 0 0 0 0 0 1 0
	0 0 1 0 1 0 0 0
	1 0 0 0 0 0 1 0
	1 0 0 0 0 0 0 0
	0 0 1 0 0 0 0 0

...	

Figure 12. Prime Blocks for Keys

3. After that, we “randomize” the key block. The key block is changed continuously as long as the plain text blocks. The changes are the character of this key substitute with the order of the key element added or subtracted by the result of mod operation of the order of key element with the length of key.
e.g. :
keyword: “KRIPTOGRAFI”
K is the eleventh alphabet. “KRIPTOGRAFI” has a length of 11 characters. Therefor the new first character in the second block is the (11 + 11 mod 11)-th alphabet which is K.

FirstKey	SecondKey

0 1 0 0 1 0 1 1
0 1 0 1 0 0 1 0
0 1 0 0 1 0 0 1
0 1 0 1 0 0 0 0
0 1 0 1 0 1 0 0
0 1 0 0 1 1 1 1
0 1 0 0 0 1 1 1
0 1 0 1 0 0 1 0
0 1 0 0 0 0 0 1
0 1 0 0 0 1 1 0
0 1 0 0 1 0 0 1
	0 1 0 1 0 1 0 1
	0 1 0 0 1 1 0 0
	0 1 0 1 0 0 0 1
	0 1 0 0 1 1 0 0
	0 1 0 0 0 0 1 0
	0 1 0 0 1 1 0 0
	0 1 0 0 1 1 0 1
	0 1 0 0 1 1 0 0
	0 1 0 0 0 0 0 1
	0 1 0 0 0 0 0 1
	0 1 0 1 0 0 0 1

Figure 13. “Randomization” Key for Substitution

After the substitution, we use Feistel network. The steps are:

1. For each block, divide it into two parts, left block and right block.
2. Generate the key with the randomize method that was told before. Randomize the key as much as the number of the blocks, so each block has different key.
3. For each randomized key, divide it too into two parts, left key and right key.
4. Then, start the Feistel network algorithm. First, for each block, apply the Feistel network function to right block. In Geldy cipher, the Feistel function is doing XNOR operation on right block with the randomized key.
5. After that, apply the XOR operation on left block with the modified right block that resulted from the feistel function (XNOR operation) before.
6. Combine the modified right block and the modified left block in exchanged position. Put the right block on the left side and the left block on the right side.
7. Finally, we get the block cipher that resulted from Geldy algorithm.

B. *Decipher Technique*

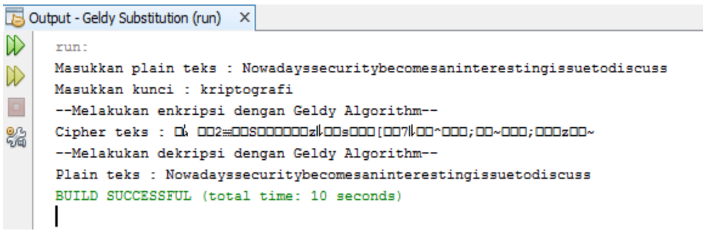
The ciphertext and key are required in decrypting process. Generally, the decipher technique of Geldy cipher is reversing the Feistel rounds. After that, reversing the substitution phase (change back the key with mod operation, change back the plaintext by XOR operation and shifted reversely the element of blocks).

IV. Analysis

Geldy cipher is quite strong and safe. It is because Geldy cipher can not be attacked with three most common used attack: ciphertext-only attack, known-plaintext attack, and chosen-plaintext attack. In ciphertext-only attack, the attacker can not break the Geldy cipher, because the key in Geldy cipher always be randomized. Since ciphertext-only attack use exhaustive key search, it will not done effectively bcause of the randomized key.

With known-plaintext attack and chosen-plaintext attack, Geldy cipher is still unbreakable. It is because in Geldy cipher, the plain text is operated in many steps with different operations. Therefore, known-plaintext and chosen-plaintext is also useless. Besides analysis of the attack, we also analyze the algorithm. Since Geldy cipher algorithm randomize the blocks in some different operations and each block has different key, Geldy cipher has complexity 2^8 * length of key or O(n^2). Based on the analysis of the cryptography attack and the algorithm, it is proved that Geldy cipher is quite strong and safe.

Here is an example of encryption and decryption using Geldy cipher.



```
run:
Masukkan plain teks : Nowadayssecuritybecomesaninterestingissuetodiscuss
Masukkan kunci : kriptografi
--Melakukan enkripsi dengan Geldy Algorithm--
Cipher teks : □꜀ □□2⩸□□S□□□□□□z⫴□□s□□□[□□7⫴□□^□□□;□□~□□□;□□□z□□~
--Melakukan dekripsi dengan Geldy Algorithm--
Plain teks : Nowadayssecuritybecomesaninterestingissuetodiscuss
BUILD SUCCESSFUL (total time: 10 seconds)
```

Figure 14. Geldy Cipher Testing

V. Conclusions and Future Works

Encryption is one of the important issue in computer security. It needs creativity and uniqueness to be more secured. One of the encryption algorithm in modern cryptography is block cipher. There are various kind of block cipher. In this paper, Geldy algorithm : a new modification of block cipher is proposed.

Geldy cipher concept is more based on substitution and Feistel network diagram. In experiment, it is proven that Geldy is quite strong in security analysis. Beside of the key is changed every cycle, the plain blocks is also changed every cycle with shift and mod operations. In addition, Feistel network also make the encryption more complex. It is because Feistel network divide each block into two parts, then each part applied with different operation, and the two parts combined in exchanged position.

In future, this algorithm can be researched more comprehensively. Geldy is also can be customized and improved in both substitution or Feistel based network diagram phase. We hope this algorithm can contribute the study of crypthography especially in block cipher.

Acknowledgement

First of all, Authors would like to thank God for His blessing and grace Authors can finish this paper. Authors also want to express their thanks to Cryptography Lecturer, Dr. Ir Rinaldi Munir, M.T., whose has given advices and assistance. Besides that, the completion of this paper could not have been possible without the participation and assistance of so many people whose names may not all be enumerated. Therefore, Authors would like to thank them for their contributions which are sincerely appreciated and gratefully acknowledged.

References

[1] A. Menezes, P. vanOorschot, and S. Vanstone, “Handbook of Applied Cryptography”, Chapter 7, Block Cipher, 1996, CRC Press.

[2] Bellare. Mihir, “Introduction to Modern Cryptography” University of California San Diego, Chapter 1, *https://cseweb.ucsd.edu/~mihir/cse207/s-intro.pdf.* accessed on 21 Maret 2016 13:54.

[3] Bellare. Mihir, “Block Ciphers” University of California San Diego, Chapter 2, *https://cseweb.ucsd.edu/~mihir/cse207/s-bc.pdf.* accessed on 21 Maret 2016 13:54.

[4] Thawte, Inc. 2013. “History of Cryptography, An Easy to Understand History of Cryptography”, United States, http://book.itep.ru/depository/crypto/Cryptography_history.pdf. accessed on 21 Maret 2016 21.27.

[5] http://www.tutorialspoint.com/cryptography/modern_cryptography.htm, accessed on 23 Maret 2016 22.33.