Enhancing Data Security of Columnar Transposition Cipher by Fibonacci Codes Algorithm

Saidi Ramadan Siregar[1], F Fadlina[2], Surya Darma Nasution[1]
{saidiramadan89@gmail.com, Fadlinamkom11@gmail.com, darmashadow@gmail.com}

[1]Department of Computer Science, STMIK Budi Darma, Medan, Indonesia
[2]Department of Computer Science, AMIK STIEKOM, Sumatera Utara, Indonesia

Abstract. Many people want their data unknown to others. To maintain the security and confidentiality of a data, one can apply the cryptographic algorithm. Although many cryptographic algorithms exist today, many have been solved by cryptanalysis. To improve the security of a text data, can by adding with the compression algorithm. Although the cryptographic algorithm used is a classical cryptographic algorithm, if added to the compression algorithm it will complicate cryptanalysis to find the original data. This research is done by applying a columnar transposition algorithm which is a simple classical cryptography algorithm, with Fibonacci Codes Algorithm to compress data. The results obtained are more secure data security than using the modern cryptographic algorithm, because the character results from the encryption process changed again after the encoding process and the number of characters ciphertext fewer than the number of plaintext characters.

Keywords: Data Security, Cryptography, Compression, Columnar Transposition Cipher, Fibonacci Codes.

1 Introduction

The need for data security is very important today, and many ways have been done to improve the security of data. One way to improve security is to apply cryptography algorithms to encrypt data[1]–[3]. There are also implementing compression algorithms to help improve the security of a data. So once the data is encrypted, it will be encoded using a compression algorithm. In the previous research that has been done by combining two cryptographic algorithms namely vigenère cipher and columnar transposition cipher concluded that there are a large deviation and variation from normal observation and can indicate very high strength because of its hybrid nature [4]. Other studies were also conducted to improve data security on cipher transposition columnar using caesar cipher and rail fence cipher[5]. Research by applying a compression algorithm[2] to strengthen the security of cryptographic algorithm has been done before by applying goldbach codes algorithm to compress the encryption of vigenere cipher[6][7].

WMA 2019, December 11-14, Medan, Indonesia

DOI 10.4108/eai.11-12-2019.2290839

2 Theory

2.2 Cryptography

Security aspects of a data, information, or secret messages cannot be separated from cryptography[3], [8], [9]. But not all security aspects of a data or information can be solved by cryptography. Cryptography can be interpreted as a science to secure a message or information. Cryptography consists of classical cryptographic algorithms and modern cryptographic algorithms. In this research, a classical cryptographic algorithm will be applied[10][11].

2.2 Data Compression

Data compression is a process for encoding information using smaller bits smaller than its unencoded original data. Data compression technique there are 2 (two) that is lossy compression technique and lossless compression technique. Lossy compression technique there are some pieces of information lost in the data at the time the compression process occurs. While the compression technique is lossless the opposite of the lossy compression technique that is the absence of information lost during the compression process occurs[12][13].There are some important factors in compression, Time Process (time required to compress), Completeness (Completeness of data after the compression process), Compression Ratio (data size after compression), Optimally (comparison between file size after compression process is the same or not with the file before compressed)[14][7].

2.3 Columnar Transposition Cipher

Columnar Transposition Cipher is a classic cryptographic algorithm that is simple and very easy to implement. Although relatively easy to solve algorithm, but if combined with other algorithms it will be difficult to solve it [5]. In columnar transpositions, messages are written in sequences of fixed length, and then read them again column by column, and the columns are selected in some scrambled order. Both the row width and the column permutation are usually determined by a keyword. For example, the word “KERTAS” with a length of 6 characters (so the line with length 6), and permutations are defined in alphabetical order of letters in the keyword. In this case, the sequence is "3 2 4 6 1 5"[4][15][16].

2.4 Fibonacci Codes

Fibonacci codes are one of the compression algorithms called variable length code, where smaller integers or frequencies that appear more often will get a short code. The code ends with two bits one, and the value obtained is the sum of the corresponding Fibonacci values for the specified bit (except the last bit, which is the end of the code)[17][14].
The steps of forming a codeword are as follows:

a. Find a positive integer n that is greater than or equal to 2.
b. Find the largest fibonacci number f smaller or equal to n, subtract the value of n by f and record the remaining n-value reduction by f.
c. If the deduced number is a number found in the Fibonacci sequence F (i), add the number "1" to i-2 in the Fibonacci codes to be formed.
d. Repeat step 2, exchange the value of n with the remaining n value reductions by f until the rest of the n-value reduction with f is 0.
e. Add the number "1" to the far right of the Fibonacci codes to be established.

To decode a Fibonacci codes, remove the rightmost "1" number, then substitute and add the remaining Fibonacci codes using the Fibonacci sequence.

3 Result And Discussion

For the process of securing text data by applying columnar transposition and Fibonacci codes algorithm consists of 4 (four) processes are:

3.1 Encryption With Columnar Transposition

For example, the text data to be secured is "WE EXECUTE HIM THIS AFTERNOON" and the key is the word "GAPLEK", then by using columnar transposition then the process happens.

Table 1: Sorting Initial Character

G	A	P	L	E	K
W	E		E	X	E
C	U	T	E		H
I	M		T	H	I
S		A	F	T	E
R	N	O	O	N	X

If there is an empty field at the end, it will be filled with the character "X". And after reordered, it will get the following results:

Table 2: Character After Ordered

A	E	G	K	L	P
E	X	W	E	E	
U		C	H	E	T
M	H	I	I	T	
	T	S	E	F	A
N	N	R	X	O	O

Read the message column by column, then the resulting ciphertext is ”EXWEE U CHETMHIIT TSEFANNRXOO”.

3.2 Encoding With Fibonacci Codes

The result of the encryption process of columnar transposition is ”EXWEE U CHETMHIIT TSEFANNRXOO” will be compressed using Fibonacci Code. Before compression is done, first count the number of bits of the text data. before compressed can be seen in table 1.

Table 3. Data Before Compression

No	Char	Dec	Binary	Freq	Bit	Bit x Freq
1	E	69	01000101	5	8	40
2	X	88	01011000	2	8	16
3	W	87	01010111	1	8	8
4	Sp	83	01010011	4	8	32
5	U	85	01010101	1	8	8
6	C	67	01000011	1	8	8

No	Char	Dec	Binary	Freq	Bit	Bit x Freq
7	H	72	01001000	2	8	16
8	T	84	01010100	3	8	24
9	M	77	01001101	1	8	8
10	I	73	01001001	2	8	16
11	S	83	01010011	1	8	8
12	F	70	01000110	1	8	8
13	A	65	01000001	1	8	8
14	N	78	01001110	2	8	16
15	R	82	01010010	1	8	8
16	O	79	01001111	2	8	16
Jumlah						240

It can be seen from table 1 that the amount of encrypted text data is 240 bits or 30 bytes. The encoding process is done by sorting the data in table 1 in accordance with the more frequencies positioned above. Once sorted, then use the fibonacci code to generate the codeword of each character. The process can be seen in table 2.

Table 4. Codeword Formation

No	Char	Freq	Codeword	Bit	Bit x Freq
1	E	5	11	2	10
2	Sp	4	011	3	12
3	T	3	0011	4	12
4	X	2	1011	4	8
5	H	2	00011	5	10
6	I	2	10011	5	10
7	N	2	01011	5	10
8	O	2	000011	6	12
9	W	1	100011	6	6
10	U	1	010011	6	6
11	C	1	001011	6	6
12	M	1	101011	6	6
13	S	1	0000011	7	7
14	F	1	1000011	7	7
15	A	1	0100011	7	7
16	R	1	0010011	7	7
Jumlah					136

From table 2 can be seen the result of encoding process using fibonacci code is 136 bits or 17 bytes. The next step is to replace the character of the ciphertext in accordance with the codeword that has been generated.

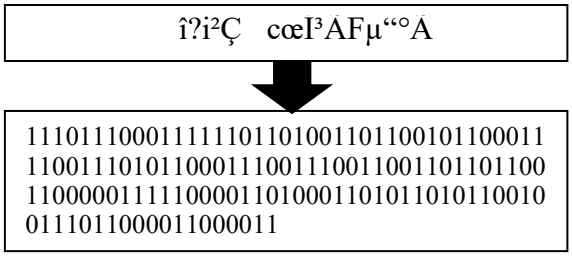
E 11	X 1011	W 100011	E 11	E 11
Sp 011	U 010011	Sp 011	C 001011	H 00011
E 11	T 0011	M 101011	H 00011	I 10011
I	T	Sp	Sp	T

10011	0011	011	011	0011
S 0000011	E 11	F 1000011	A 0100011	N 01011
N 01011	R 0010011	X 1011	O 000011	O 000011

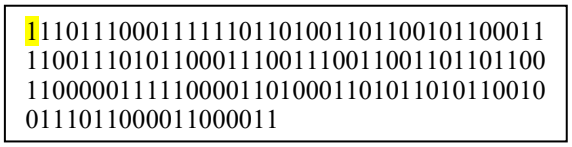
After each character is replaced with each codeword, then the codeword will be sorted and divided into 8 bits and then converted into characters. The result of the process is **î?i²Ç cœÍ³ÃFµ“°Ã**.

3.3 Decoding With Fibonacci Codes

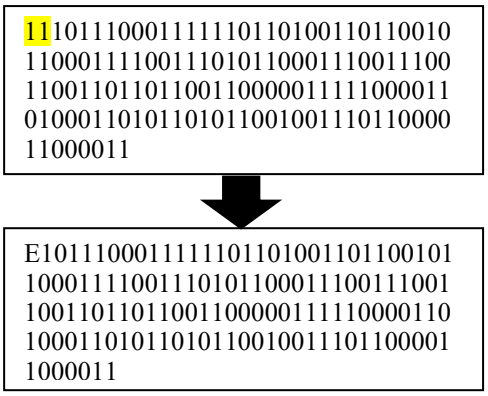
Decoding is done to convert compressed text into ciphertext. The first step of the decoding process is to change the sentence “**î?i²Ç cœÍ³ÃFµ“°Ã**” into a binary number.



Do a string reading of the index 1 to the n index. Index-1 = "1" and not on the codeword table.



Because it is not in the codeword table then it is added with the 2nd index to "11" and it is in the codeword table that is the character "E".



Followed by reading the string of bits from the 3rd index = "1", and not on the codeword table.

E101110001111110110100110110010
1000111100111010110001110011100
1
1001101101100110000011111000011
0
1000110101101011001001110110000
1
1000011

Because it is not in the cordword table add the 4th index = "0" to "10", and not on the codeword table.

E101110001111110110100110110010
1000111100111010110001110011100
1
1001101101100110000011111000011
0
1000110101101011001001110110000
1
1000011

Because it is not in the cordword table add the 5th index = "1" to "101", and not on the codeword table.

E1011110001111110110100110110010
1000111100111010110001110011100
1
1001101101100110000011111000011
0
1000110101101011001001110110000
1
1000011

Because it is not in the cordword table add the 6th index= "1" to "1011", and there in cordword table that is character "X".

E10111000111111011010011011001
0
1000111100111010110001110011100
1
1001101101100110000011111000011
0
1000110101101011001001110110000
1
1000011

EX1000111111011010011011001010
0
0111100111010110001110011100
1
1101101100110000011111000011
0
0110101101011001001110110000
1
0011

Followed by reading the bit string of the 7th index = "1", and not on the codeword table.

EX1000111111011010011011001010
0
0111100111010110001110011100
1
1101101100110000011111000011
0
0110101101011001001110110000
1
0011

Because it is not in the cordword table add the 8th index= "0" to "10", and not on the codeword table.

EX10001111110110100110110010110001111001110101100011100111001100110110011000000111110000110100011010110101100100111011001100001100 0011

Because it is not in the cordword table add the 9th index= "0" to "100", and not on the codeword table.

EX10001111110110100110110010110001111001110101100011100111001100110110011000000111110000110100011010110101100100111011001100001100 0011

Because it is not in the cordword table add the 10th index= "0" to "1000", and not on the codeword table.

EX10001111110110100110110010110001111001110101100011100111001100110110011000000111110000110100011010110101100100111011001100001100 0011

Because it is not in the cordword table add the 11th index= "1" to "10001", and not on the codeword table.

EX10001111110110100110110010110001111001110101100011100111001100110110011000000111110000110100011010110101100100111011001100001100 0011

Because it is not in the cordword table add the 12th index= "1" to "100011", and there in cordword table that is character "W".

EX100011111101101001101100101100
011110011101011000111001110011001
110110110011000001111100001101000
011010110101100100111011000011000
0011

EXW111101101001101100101100011
100111010110001110011100110011011
101100110000011111000011010001100
101101011001001110110000110000011

Followed by reading the bit string of the 13th index = "1", and not on the codeword table.

EXW111101101001101100101100011
100111010110001110011100110011011
101100110000011111000011010001100
101101011001001110110000110000011

Because it is not in the codeword table then it is added with the 14th index to "11" and it is in the codeword table that is the character "E".

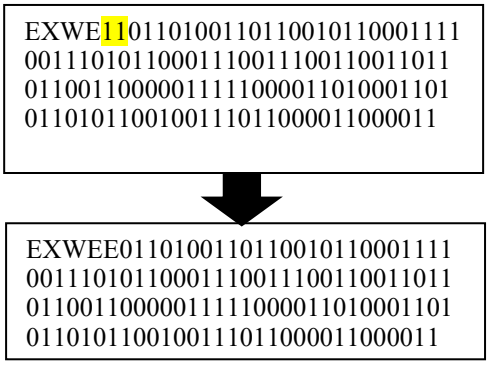
EXW1111011010011011001011000111
100111010110001110011100110011011
101100110000011111000011010001100
101101011001001110110000110000011

EXWE11011010011011001011000111
100111010110001110011100110011011
011001100000111110000110100011011
011010110010011101100001100000011

Followed by reading the bit string of the 15th index = "1", and not on the codeword table.

EXWE11011010011011001011000111
100111010110001110011100110011011
011001100000111110000110100011011
011010110010011101100001100000011

Because it is not in the codeword table then it is added with the 16th index to "11" and it is in the codeword table that is the character "E".



If it continues until the last index, it will generate a sentence “EXWEE U CHETMHIIT TSEFANNRXOO”

3.4 Decryption With Columnar Transposition

Decryption is done to change the text of the decoding results into plaintext or original text. Steps of the decryption process are:

a. Enter the key, in which case the key is "gaplek". Sort key characters from the beginning of the alphabet letter to the end, making it "aegklp". Arrange ciphertext and input on the available fields. And the result is as follows:

A	E	G	K	L	P
E	X	W	E	E	
U		C	H	E	T
M	H	I	I	T	
	T	S	E	F	A
N	N	R	X	O	O

b. Sort the key position as before and reread the character from beginning to end.

G	A	P	L	E	K
W	E		E	X	E
C	U	T	E		H
I	M		T	H	I
S		A	F	T	E
R	N	O	O	N	X

c. Eliminate the character "X" at the end of the sentence, so the result back to the initial text is "WE EXECUTE HIM THIS AFTERNOON"

4 Conclusions

Increasing the security of data performed produces more character variation, the number of characters on the ciphertext is smaller than the number of characters in plaintext. As a result

of these security improvements, even its algorithmic powers are like modern cryptographic algorithms, and it will be difficult to break the original text.

References

[1] R. Rahim, "Man-in-the-middle-attack prevention using interlock protocol method," *ARPN J. Eng. Appl. Sci.*, vol. 12, no. 22, pp. 6483–6487, 2017.

[2] R. Rahim, M. Dahria, M. Syahril, and B. Anwar, "Combination of the Blowfish and Lempel-Ziv-Welch algorithms for text compression," *World Trans. Eng. Technol. Educ.*, vol. 15, no. 3, pp. 292–297, 2017.

[3] H. Nurdiyanto, R. Rahim, and N. Wulan, "Symmetric Stream Cipher using Triple Transposition Key Method and Base64 Algorithm for Security Improvement," *J. Phys. Conf. Ser.*, vol. 930, no. 1, p. 012005, Dec. 2017.

[4] Q.-A. Kester, "A HYBRID CRYPTOSYSTEM BASED ON VIGENÈRE CIPHER AND COLUMNAR TRANSPOSITION CIPHER," *Int. J. Adv. Technol. Eng. Res.*, vol. 3, no. 1, pp. 141–147, 2013.

[5] J. A. Dar, "ENHANCING THE DATA SECURITY OF SIMPLE COLUMNAR TRANSPOSITION CIPHER BY CAESAR CIPHER AND RAIL FENCE CIPHER TECHNIQUE .," vol. 4, no. 2, pp. 1054–1061, 2014.

[6] S. D. Nasution, G. L. Ginting, M. Syahrizal, and R. Rahim, "Data Security Using Vigenere Cipher and Goldbach Codes Algorithm," *Int. J. Eng. Res. Technol.*, vol. 6, no. 01, pp. 360–363, 2017.

[7] S. D. Nasution and Mesran, "Goldbach Codes Algorithm for Text Compression," *IJournals Int. J. Softw. Hardw. Res. Eng*, vol. 4, no. December, pp. 43–46, 2016.

[8] R. Rahim and A. Ikhwan, "Study of Three Pass Protocol on Data Security," *Int. J. Sci. Res.*, vol. 5, no. 11, pp. 102–104, Nov. 2016.

[9] R. Rahim and A. Ikhwan, "Cryptography Technique with Modular Multiplication Block Cipher and Playfair Cipher," *Int. J. Sci. Res. Sci. Technol.*, vol. 2, no. 6, pp. 71–78, 2016.

[10] S. D. Nasution, "Penerapan Metode Linier Kongruendan Algoritma Vigenère Chiper Pada Aplikasi Sistem Ujian Berbasis Lan," *Pelita Inform.*, vol. 4, no. 1, pp. 94–102, 2013.

[11] W. Stallings, "Cryptography and Network Security Principles and Practices," 4th Editio., .

[12] M. R. Irliansyah, S. D. Nasution, and K. Ulfa, "Penerapan Metode Deflate Dan Algoritma Goldbach Codes Dalam Kompresi File Teks," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 1, no. 1, pp. 186–189, 2017.

[13] A. H. Lubis, S. D. Nasution, and K. Ulfa, "Penerapan Algoritma Goldbach Codes Dalam Pemampatan Short Message Service Berbasis Android," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 1, no. 1, pp. 171–175, 2017.

[14] D. Salomon and G. Motta, *HandBook Of Data Compression*. Springer.

[15] M. B. Pramanik, "Implementation of Cryptography Technique using Columnar Transposition," *Int. J. Comput. Appl.*, pp. 19–23, 2014.

[16] R. Sadikin, *Kriptografi Untuk Keamanan Jaringan*. Yogyakarta: Penerbit Andi, 2012.

[17] D. Salomon, *Data Compression The Complete Reference FourthEdition*, vol. 53, no. 9. Springer, 2007.