Data Security Using Vigenere Cipher and Goldbach Codes Algorithm

Surya Darma Nasution[1]
Departement of Computer Engineering
STMIK Budi Darma
Medan, Indonesia
Jl. Sisingamangaraja XII No. 338,
Siti Rejo I, Medan Kota, Kota Medan,
Sumatera Utara, 20216

Guidio Leonarde Ginting[2]
Departement of Computer Engineering
STMIK Budi Darma
Medan, Indonesia
Jl. Sisingamangaraja XII No. 338,
Siti Rejo I, Medan Kota, Kota Medan,
Sumatera Utara, 20216

Muhammad Syahrizal[3]
Departement of Computer Engineering
STMIK Budi Darma
Medan, Indonesia
Jl. Sisingamangaraja XII No. 338,
Siti Rejo I, Medan Kota, Kota Medan,
Sumatera Utara, 20216

Robbi Rahim[4]
Departement of Computer Engineering
Medan Institute of Technology
Medan, Indonesia
Jl. Gedung Arca No.52 Kota Medan,
Sumatera Utara,

***Abstract*— Vigenere chipper is one standard cryptographic algorithm, this algorithm very simple to use substitution as in Caesar cipher to encode the message text. One disadvantage of an extended vigenere key cipher which can determine by using a method, the method kasiski. in vigenere cipher, phrases 4/E: in the ciphertext generated from the encryption process so that the method kasiski can determine the encoded text. Goldbach codes algorithm is an algorithm for compression, which will be used to address the weaknesses in vigenere cipher. In this paper will try to discuss security with vigenere ciphertext, then ciphertext produced will be processed again using algorithms Goldbach codes. By applying the algorithm, Goldbach codes the result from the course of securing data using vigenere Chipper becomes harder to guess the original text despite using methods kasiski due to be acquired is the message of a different character.**

Keywords*— *Vigenere Cipher, Goldbach Codes Algorithm, Cryptography, Compression.

I. INTRODUCTION

There are many aspects to security and applications, ranging from secure commerce and payments to private communications and protecting passwords. Cryptography is an essential part of secure communications [1]. The crucial goal of cryptography is to hide information from unauthorized individuals; most algorithms can break, and the information can be exposed if the attacker has enough time, desire, and resources [2].

In the previous research has enhanced security with vigenere cipher change the formula so that it eliminates the possibility for a review kasiski method decode and Predict patterns. In traditional Vigenere technique, the plaintext considered as a sequence of alphabets without any space between them. It may create a problem for the receiver to read the message by inserting spaces between words and receiver needs to guess the exact place to add space in decrypted plaintext. In proposed technique, they eliminate this problem by introducing a different numeric value for space in each table [2].

The previous research was purpose by Phillip I Wilson and Mario Garcia. By adding a few bits of random padding to each byte, one can diffuse the statistical retentiveness found within most messages. A one-way function will determine the exact quantity of pad to eliminate the distinguishable of the message bits from the padded random bits. This methodology moderately increases the size of the ciphertext, but significantly enhances the security of the cipher [3].

II. THEORY

A. Vigenere Cipher

The Vigenere cipher is a method of encrypting alphabetic text by using a series of different Caesar ciphers based on the letters of a keyword. It is a simple form of polyalphabetic substitution [4] [5]. The Cipher spoils the statistics of a simple Caesar cipher by using multiple Caesar ciphers. The technique named after its inventor, Blaise de Vigenere from the court of Henry III of France in the sixteenth century, and was considered unbreakable for some 300 years [6] [7].

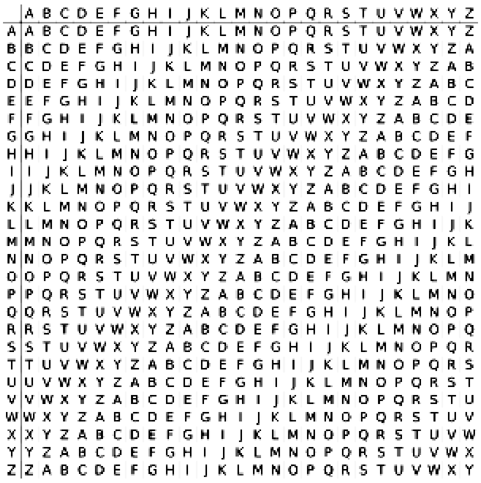
Vigenere can also view algebraically. If the letters A–Z are taken to be the numbers 0–25, and addition perform modulo 26, then Vigenere encryption E using the key K can be written,

$$Ci = EK\ (Pi) = (Pi + Ki) \bmod 26 \qquad (1)$$

and decryption D using the key K,

$$Pi = DK(Ci) = (Ci - Ki) \bmod 26 \qquad (2)$$

Vigenere cipher process could also use a table, as in Figure 1 below:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Fig 1. The Vigenere Square

B. *Goldbach Codes Algorithm*

Lossy data compression contrast with lossless data compression. Lossy data compression algorithms do not produce an exact copy of the information after decompression as was present before compression. In these schemes, some loss of information is acceptable. Lossy Compression reduces the file size by eliminating some redundant data that won't be recognized by humans after decoding [8] [9].

In 2001, Peter Fenwick had the idea of using the Goldbach conjecture (assuming that it is true) to design an entirely new class of codes, based on prime numbers. The prime numbers can serve as the basis of a number system, so if we write an even integer in this scheme, its representation will have exactly two 1's. Thus, the even number 20 equals 7 + 13 and can, therefore, until write 10100, where the five bits are assigned the prime weights (from left to right) 13, 11, 7, 5, and 3. Now reverse this bit pattern so that its least-significant bit becomes 1, to yield 00101. Such a number is easy to read and extract from a long bitstring. Only stop reading at the second 1. Recall that a similar rule reads the unary code (a sequence of zeros terminated by a single 1): stop at the first 1. Thus, the Goldbach codes can be considered an extension of the simple unary code [10].

Goldbach's original conjecture (sometimes called the "ternary" Goldbach conjecture), written in a June 7, 1742, letter to Euler, states "at least it seems that every integer that is greater than 2 is the sum of three primes". This makes sense because Gold Bach considered 1 a prime, a convention that longer followed. Today, mod earn his statement would be rephrased to "every integer greater than 5 is the sum of three primes" [11].

Goldbach codes algorithm implementation to compress text data, with the following steps:

1. Read all the characters in the message to calculate the frequency of occurrence of each character, where the character who appears most is in the first (n = 1) and so on.
2. Find a codeword for each character to be encoded, by finding prime numbers which can represent the sum of two primes. A prime number starting from the number 3, 5, 7 and so on.
3. Codeword only has two "1". Search primes represents the sum stops at "1" the second one, for example: Even number = 24, and 11 + 13 = 24 True. After getting a couple of primes are correct, then the bit pattern that obtains is 11000. Then the bit pattern is reversed to the most rear bit to "1", resulting in a pattern of bits 00011.

TABLE I.
The Goldbach G0 Codes

n	2(n+3)	Primes	Codeword
1	8	3+5	11
2	10	3+7	101
3	12	5+7	011
4	14	3+11	1001
5	16	5+11	0101
6	18	7+11	0011
7	20	7+13	00101
8	22	5+17	010001
9	24	11+13	00011
10	26	7+19	0010001
11	28	11+17	000101
12	30	13+17	000011
13	32	13+19	0000101
14	34	11+23	00010001

III. RESULT & DISCUSSION

In this section, a process that occurs to secure text data is to encrypt it using vigenere cipher and then to compress the resulting ciphertext. Here is the process that happens:

Plaintext : GOOD MORNING
Key : SURYA

If the key length is smaller than the number of characters in plaintext, the key characters will be repeated as many as the number of characters in plaintext.

Plaintext : GOOD MORNING
Key : SURY ASURYAS

The process occurs as follows :
Ci = EK (Pi) = (Pi + Ki) mod 26

G and S
Ci = (6+18) mod 26
= 24 mod 26
= 24 (Y)

O and U
Ci = (14+20) mod 26
= 34 mod 26
= 8 (I)

O and R
Ci = (14+17) mod 26
= 31 mod 26
= 5 (F)

If the vigenere cipher process continues until all plaintext is encrypted, then the resulting ciphertext is "YIFB MGLEGNY". Having obtained the results of vigenere cipher, then the process will be seen by compressing the ciphertext using Goldbach codes.

TABLE II.
Compression Process Ciphertext

Char	Freq	n	2(n+3)	Primes	Codeword
Y	2	1	8	3+5	11
G	2	2	10	3+7	101
I	1	3	12	5+7	011
Space	1	4	14	3+11	1001
F	1	5	16	5+11	0101
B	1	6	18	7+11	0011
M	1	7	20	7+13	00101
L	1	8	22	5+17	010001
E	1	9	24	11+13	00011
N	1	10	26	7+19	0010001

From Table 2 it ciphertext "YIFB MGLEGNY" will turn into bits as follows: "11 011 0101 0011 1001 00101 101 010001 00011 101 0010001 11". From these bits will be split by eight bits and converted into ASCII characters so that it will produce a different shape. And the process can be seen in Table 3 below:

TABLE III.
Compression Results Using Goldbach Codes

Biner	Ascii	Char
11011010	218	Ú
10011100	156	Œ
10010110	150	–
10100010	162	¢
00111010	58	:
01000111	71	G

From Table 3 it can be seen that the result of compression using codes Goldbach produces "ÚŒ–¢: G", to restore the encoding result into plaintext initial compression, the process is needed as:

1. Decompression using Goldbach codes algorithm. The results decompression Goldbach codes algorithm is ciphertext.
2. The phrase "¢ ÚŒ-: G" will be in the decompression, the process is to transform into a binary.

TABLE IV.
Decompression Using Goldbach Codes

Char	Ascii	Biner
Ú	218	11011010
Œ	156	10011100
–	150	10010110
¢	162	10100010
:	58	00111010
G	71	01000111

3. The binary results are “110110101001110010010110 1010001000111101001000111”
4. From the binary values will decompress by looking at the structure in Table 2.
5. Results of decompression is "YIFB MGLEGNY"
6. Do the decryption process using a vigenere cipher.
7. Use Key “SURYA” to decrypted.

Ciphertext : YIFB MGLEGNY
Key : SURY ASURYAS

The process occurs as follows:
Pi = DK(Ci) = (Ci - Ki) mod 26

Y and S
Pi = (24-18) mod 26
= 6 mod 26
= 6 (G)

I and U
Pi = (8-20) mod 26
= 14 mod 26
= 14 (O)

F and R
Pi = (5-17) mod 26
= 14 mod 26
= 14 (O)

If the vigenere cipher process continues until all ciphertext is decrypted, then the resulting plaintext is "GOOD MORNING".

IV. CONCLUSION

Conclusions from the course of securing data using Goldbach codes algorithm and vigenere ciphers, namely, ciphertext that has been compressed text initially be difficult to predict despite using methods kasiski. Plaintext prediction becomes difficult also due, the amount of code compression results different from the number of ciphertext characters of the original text.

V. REFERENCES

[1] S. William and W. Stallings, Cryptography and Network Security, 4/E : Pearson Education India, 2006.

[2] A. A. Soofi, I. Riaz and U. Rasheed, "An Enhanced Vigenere Cipher For Data Security," *International Journal Of Scientific & Technology Research*, vol. 5, no. 03, pp. 141-145, 2016.

[3] P. I. Wilson and M. Garcia, "A Modified Version of the Vigenère Algorithm," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 6, no. 3, pp. 140-143, 2006.

[4] Bruen, A. A. &. Forcinito and M. A, Cryptography,Information Theory, and Error-Correction: A Handbook for the 21st Century, John Wiley & Sons, 2011, p. 21.

[5] Martin and K. M, Everyday Cryptography, Oxford University, 2012.

[6] Wobst and Reinhard, Cryptology Unlocked, Wiley, 2001.

[7] Quist and A. Kester, "A cryptosystem based on Vigenère cipher with varying key," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 1, no. 10, pp. 108-113, 2012.

[8] H. Jani and J. Trivedi, "A Survey on Different Compression Techniques Algorithm for Data Compression," *International Journal of Advanced Research in Computer Science & Technology (IJARCST),* vol. 2, no. 3, 2014.

[9] A. V. Singh and G. Singh, "A Survey on Different text Data Compression Techniques," *International Journal of Science and Research (IJSR),* vol. 3, no. 7, pp. 1999-2002, 2014.

[10] S. D. Nasution and Mesran, "Goldbach Codes Algorithm For Text Compression," *International Journal of Software & Hardware Research in Engineering (IJSHRE),* vol. 4, no. 11, pp. 43-46, 2016.

[11] D. Salomon and G. Motta, "Handbook Of Data Compression Fifth Edition," Springer, 2010.