Global Scientific JOURNALS

GSJ: Volume 12, Issue 4, April 2024, Online: ISSN 2320-9186

www.globalscientificjournal.com

SUPER ENCRYPTION TECHNIQUE USING CRYPTOGRAPHY BASED ON COMBINING OF VIGENERE CIPHER AND AFFINE CIPHER FOR PRE-PAID WATER METER TOKEN

Arif Hermawan

Semarang State of Polytechnic, Faculty of Electrical Engineering, Indonesia

Email: arriff.hermawan1@gmail.com

Abstract:

Delays in payment for the use of subscription water or errors reading in reading water meter by the recorder are the main problems faced by water providers. Overcoming this, the prepaid method was developed using token system.

In this paper a methodology for securing data and messages from different kind of cryptography system, symmetric and asymmetric with various encryption methods will be elaborated.

The purpose of this research is to encrypt messages with Vigenere and Affine cipher method as a substitution cryptographic system on the water meter using the Matlab programming as a comparison toward manual calculation using table. Cryptographic system is a security system with the concept of making data into a password that is not everyone can read it. These combination cryptographic Vigenere cipher and Affine cipher method will be applied to a smart water token meter system.

Keywords: *Token*, *Vigenere cipher*, *Affine cipher*, *Encryption, Decryption, Cryptography, Water Meter*

1. INTRODUCTION

Problems that often arise in postpaid analogue water recording systems, from the side of the water supply company, are frequent delays in payment from the customer. Meanwhile, from the customer's side, manual reading and recording of water meters by officers causes the volume of water to often not match the actual usage and customers cannot regulate water usage. To overcome the problems that occur, a prepaid water meter payment system with credit tokens has been developed. It is expected that with the credit token system, the officer's manual recording errors and water usage consumption can be controlled by customers.

Problems arise when using token-based digital water meters, namely the hacking of consumer data by irresponsible parties with the aim of personal or commercialized interests. In

this research, the author tries to find a solution to the problem of hacking water meter token data, namely by making a data security system based on cryptographic methods.

The author in this study emphasizes the use of computational models as a security parameter in the process of testing encryption and decryption of the authenticity of water meter tokens. Computational security parameters are performed by measuring input data on the basis of cryptographic schemes that determine computational complexity. Cryptographic schemes not only rely on perfect security but also rely on computational security. Cryptography is said to be secure if it is computationally un-hackable within a reasonable time span.

The strengths and weaknesses of each encryption algorithm can be known by evaluating several parameters as a comparison of the performance of the encryption algorithm used. The encryption parameters include computation time, security, memory usage, power consumption, output results, flexibility, architecture, reliability, and restrictions for information security.

2. CRYPTOGRAPHY MODERN

2.1. Symmetric Key Cryptography

Requires the same or symmetric key to encrypt or decrypt information. In other words, the sender and receiver of the message share the same key. Symmetric algorithms are often called single-key or secret-key algorithms. The encryption and decryption process with symmetric cryptography techniques is 1000 times faster than with asymmetric cryptography techniques.

Sample of symmetric algorithm such as Data Encryption Standard (DES), Triple DES, Blowfish and Advanced Encryption Standard (AES-128, AES-192, AES-256). AES algorithm is most widely used in encrypting information with several advantages of AES including security factors, cost and implementation.

2.1.1 Advanced Encryption Standard (AES)

AES is a block cipher with a block size of 128 bits. The key length can be 128, 192 or 256 bits. The encryption process involves ten rounds of 128 bit key, twelve rounds for 192 bit key, fourteen rounds for 256 bit key. The algorithm is called AES-128, AES-192 or AES-256 depending on the key size.

The AES encryption algorithm uses an iterative process called rounds, where the number of rounds depends on the length of the key used. The steps of each round include four layers, mainly byte replacement, row shift, row mix and round key addition.

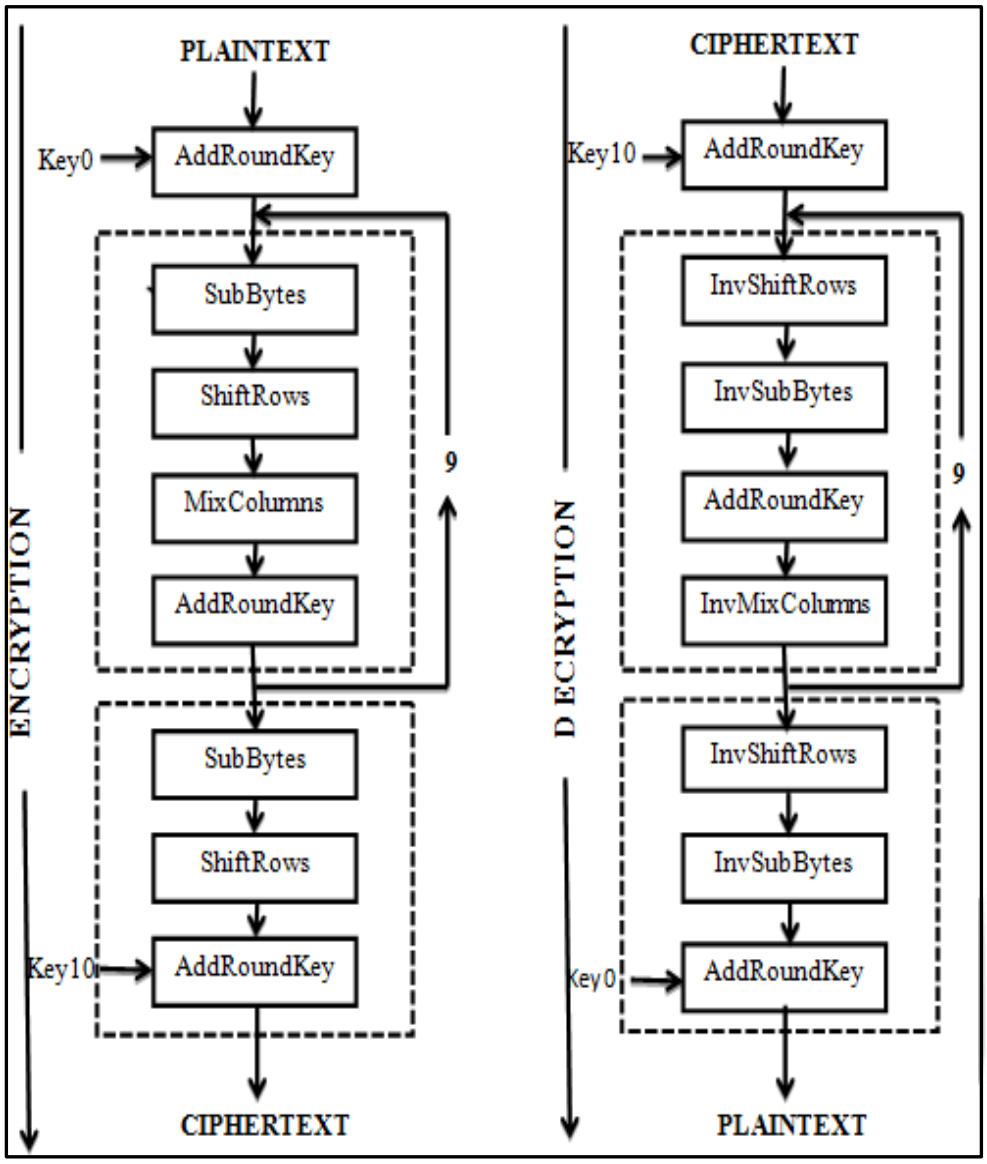


Fig.1. AES Algorithm

2.2. Asymmetric Key Cryptography

Uses private and public keys for the encryption or decryption process of information. The public key is announced to all members while the private key is kept secret by the user. The public key is used for the encryption process, so it is often called a public key algorithm. The sender uses the public key sent to the receiver to encrypt the message.

Sample of asymmetric algorithm such as Rivest Shamir Adleman (RSA), Diffie-Hellman (DH), Ellipts Curve Cryptography (ECC). The most widely used in encrypting information are RSA and ELGAMAL.

2.2.1 Rivest Shamir Adleman (RSA)

An asymmetric cryptography technique that uses two keys, a public key for encipherment and a private key for decipherment. RSA is widely used for secure communications.

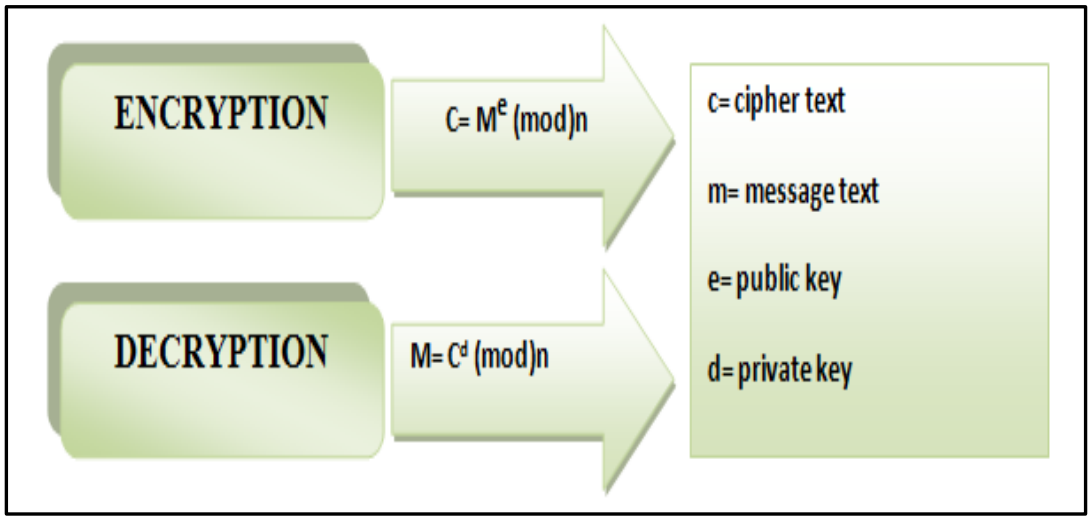


Fig.2. Encryption dan Decryption Process RSA

Cryptographic algorithms are very diverse, one of which is the substitution cryptography technique which is a process of changing the plaintext which is replaced or exchanged with characters from the alphabetical sequence which aims to increase the security of the message or information sent. In the classic cryptographic substitution technique, there are several algorithms such as Caesar cipher, Affine cipher, Hill cipher, Vigenere cipher, Engineered chip, Caster cipher, One-time pad, and other cryptographic algorithms.

3. CYRPTOGRAPHY CLASSICAL

Classical cryptographic algorithms are part of symmetric key cryptography, which is divided into two algorithms of substitution and transposition. In this research, the author only discusses the substitution algorithm. The substitution method is done by changing one letter or character in the original message according to the rules of the key and changing it into another character in the secret keyword (ciphertext).

The substitution method is part of classical cryptography which is divided into two algorithms Monoalphabetic and Polyalphabetic. Vigenere cipher method is part of the polyalphabetic algorithm. Vigenere cipher is used to encrypt alphabetic or numeric text by using a series of different Caesar ciphers based on the letters or numbers of the keyword. Affine cipher is one of the monoalphabetic substitution cryptographic techniques, and are a development of Caesar ciphers.

Vigenere cipher method is a cryptography that uses the same key for its encryption and decryption processes, otherwise known as a symmetric key. In its encryption, Vigenere cipher uses the Modulo function. A random key that has the same length as the plaintext will make this method difficult to crack.

Affine cipher method is one of the classic cryptographic techniques that exchanges monoalphabetic characters where every one character in the plaintext will be exchanged for a character according to the formula used. Combination of Affine cipher and Vigenere cipher algorithm will strengthen the encryption result with a more complex algorithm method.

In this research, a security system for pulse tokens is developed using substitution cryptography encryption techniques with a combination of Vigenere cipher algorithm with the

Matlab-based Affine cipher. Vigenere cipher algorithm has the weakness that its encryption is easy to hack because the algorithm is simple and requires development so that the algorithm is more complex and complicated. Considering this, the author tries to combine the Vigenere cipher polyalphabet algorithm technique with the Affine cipher monoalphabet algorithm technique to obtain encryption results that are complex and difficult to break.

3.1. Vigenere Cipher

Vigenere cipher is a substitution technique in which each ciphertext can have multiple plaintexts. This cryptographic technique is done in two ways, with numbers and letters. Affine cipher is a monoalphabetic substitution algorithm. It maps each alphabet with another alphabet and vice versa for the decryption process. Affine ciphers combine multiplication and addition ciphering techniques with a pair of keys. The first key is for the multiplication technique, the second for the addition technique. Both keys are in the form of numbers, which are distributed between the sender and receiver. Table 1. Shows an example of the Vigenere cipher process using numbers.

Tabel 3.1. Vigenere cipher using Numbers

A	B	C	D	E	F	G	H	I	J
0	1	2	3	4	5	6	7	8	9
K	L	M	N	O	P	Q	R	S	T
10	11	12	13	14	15	16	17	18	19
U	V	W	X	Y	Z				
20	21	22	23	24	25				

Table 3.2. Tabula Recta Table

Kunci \ Plain Text	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

3.1.1. Encryption and Decryption

At the encryption process stage, a random key is required which is generated with a random function. The keys in this study can use single keys and plural keys along with plaintext characters. Symmetric keys will be used for the encryption and decryption process.

Equation 3.1 used for encryption process:

$$C(z) = (P(z) + K(z)) \bmod 49 \quad (3.1)$$

Equation 3.2 used for decryption process:

$$P(z) = (C(z) - K(z)) \bmod 49 \quad (3.2)$$

3.2. Affine Cipher

Affine cipher is a monoalphabetic substitution algorithm. It maps each alphabet with another alphabet and vice versa for the decryption process. Affine ciphers combine multiplication and addition ciphering techniques with a pair of keys. The first key is for the multiplication technique, the second for the addition technique. Both keys are in the form of numbers, which are distributed between the sender and the receiver.

Equation 3.3 used for encryption process:

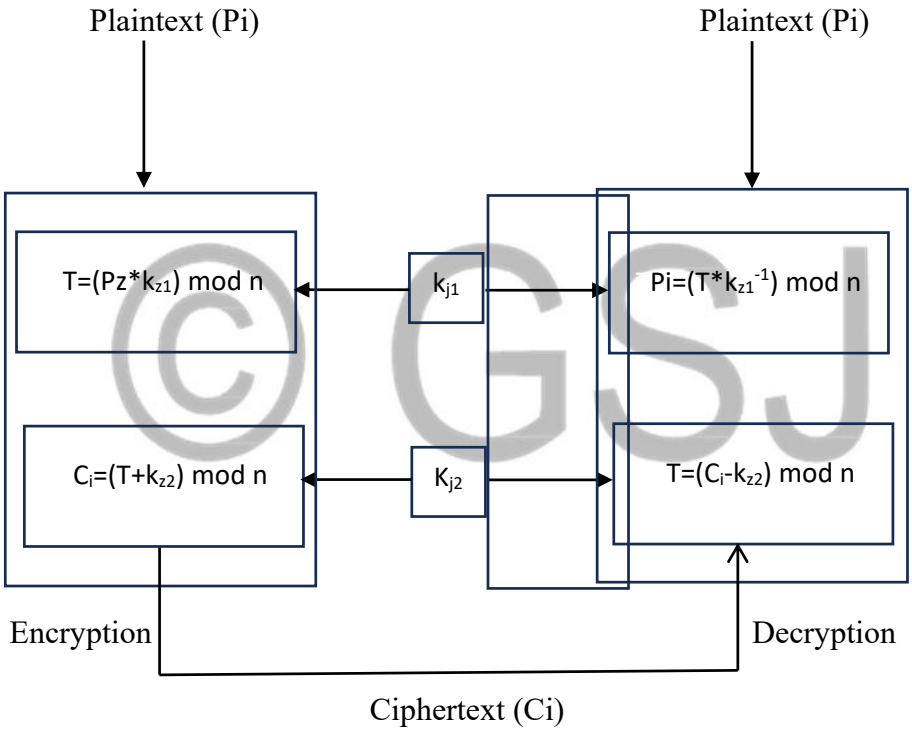
$$C(z) = (P(z) * K(z1) + K(z2)) \bmod n \quad (3.3)$$

Equation 3.4 used for decryption process:

$$P(z) = ((C(z) - K(z2)) * K(z1)^{-1}) \bmod n \quad (3.4)$$

Equation 3.5 used for Modular first key multiplier

$$K(z1)^{-1} * K(z2) \bmod 50 = 1 \quad (3.5)$$



Gambar 3.1. Encryption and Decryption Affine Cipher

Vigenere cipher has the disadvantage that the calculation of the encryption process only involves the Additive cipher which implies that this algorithm is vulnerable to hacking by analysing the frequency of the constituent letters. Vigenere algorithm also lacks diffusion and confusion properties as modern cryptography.

This research will propose encryption and decryption techniques with diffusion and confusion properties based on the concept of complex encoding by combining the Vigenere cipher algorithm with Affine cipher to improve the security of data storage and transmission on public communication networks.

The purpose of this combination of transformation techniques is to overcome the weaknesses of the Vigenere cipher algorithm, one of which is the repetition of keys following the length of the plaintext, which with cryptanalysis the cyphertext is easily hacked.

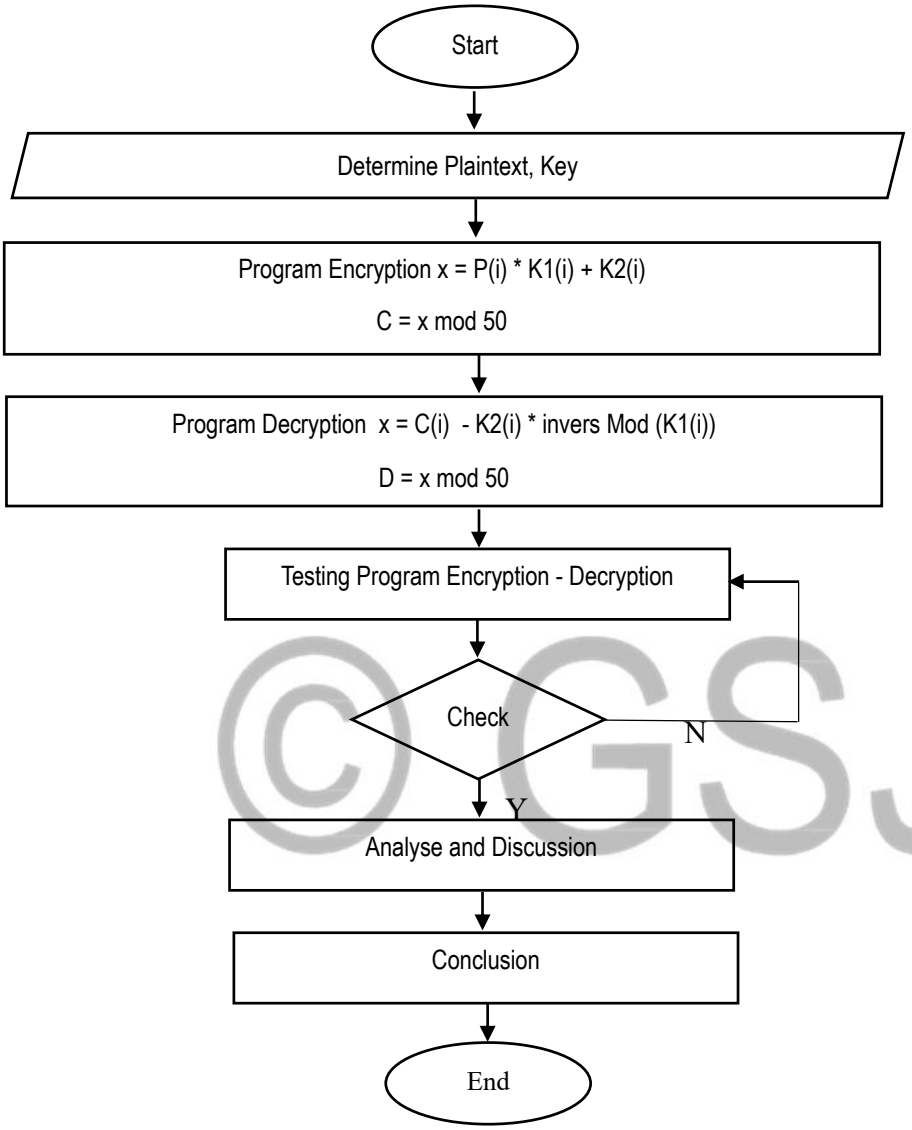


Fig. 3.2. Flowchart of Encryption – Decryption Vigenere – Affine Cipher

This research will focus on the simulation process with matlab software which aims to create and test encryption and decryption of a message, information or data into a token that will be used as a means of activating digital water meters.

3.3. Encryption and Decryption Program

Token encryption program requires some customer data as the initial information material for the process of preparing the encryption stage. Customer data such as the customer's meter identity number, purchase date, token amount and check code are required for the encryption program creation process. The process begins by entering the data into a series of matlab programs created. The next process is to perform an encryption process that functions to convert the entire code into a secret message arrangement, and into a series of characters from the credit token.

In the Vigenere-Affine cipher algorithm technique, two tables will be used, addition and multiplication, both of which are used in the encryption and decryption process that goes through two stages of the transformation process. A pair of keys is used in each encryption and decryption process, the first key and the second key. In the encryption process, the first key is used with the multiplicative *cipher* in the first stage transformation, and the second key is used with the additive *cipher* in the second stage. The first and second stage transformations are described in the formula below:

$$T = (P \times k_1) \bmod n \quad (3.6)$$

$$C = (T + k_2) \bmod n \quad (3.7)$$

The Encryption process is performed when there is a token purchase:

1. Define the original message which includes alphabetic letters, special characters, key phrases and numeric numbers.
2. Assigns the original message of 49 characters as modulo,
3. Input the original message consisting of 15 characters with an arrangement of 5 digits of alphabet or numbers for meter id, 6 digits of numbers for purchase date, 3 digits of numbers for purchase amount and 1 digit number of check code into the matlab program.
4. Assign a random key
5. Set the token amount
6. Assign a check code
7. Setting the ratio
8. Run the encryption process
9. Test the authenticity of the token obtained by comparing the test results of the *tabula recta table*.
10. If the token test result is wrong, then the encryption process is repeated until the token result is the same as the *tabula recta table* manual test result.

Token Decryption Process:

1. Read the token or data from the encryption process
2. Inputting the token data into the decryption program with the same key as during the encryption process
3. Token decryption process
4. Testing the authenticity of the token by calculating the *tabula recta table.*
5. If the token test result is wrong, then the decryption process is repeated until the token result is the same as the *tabula recta table* manual test result.

This study simulated the arrangement of characters representing customer identity data or meter number, purchase date and check code representing the nominal credit. The token code constituent is 15 characters with the configuration as shown in Table 3.3

Table 3.3. Character Composition of Token Code

ID Meter					Date Purchase						Nominal			Check
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Token arrangement was created using matlab software. The classical cryptographic encryption method of substitution, namely the polyalphabetic Vigenere cipher is used in token coding. Vigenere cipher technique is performed quickly using an alphabet table called Tabula recta. The table is composed of 26 alphabets in different rows, each alphabet shifted to the left compared to the previous alphabet according to the possible number of modulo alphabets. To increase data security in the Vigenere cipher algorithm, modulo 50 is used which is composed of a combination of *plaintext* characters, special characters, key phrase characters and numbers.

Table 3.4. Composition Modulo 50

A	B	C	D	E	F	G	H	I	J
1	2	3	4	5	6	7	8	9	10
K	L	M	N	O	P	Q	R	S	T
11	12	13	14	15	16	17	18	19	20
U	V	W	X	Y	Z	1	2	3	4
21	22	23	24	25	26	27	28	29	30
5	6	7	8	9	0	SPACE	.	,	?
31	32	33	34	35	36	37	38	39	40
!	\	*	&	$	/	;	@	#	"
41	42	43	44	45	46	47	48	49	50

Simulation cases Encryption Process Vigenere Affine

Sample cases:

1.

Initial message : 123451212123006

First key : 10

Second key : 4

Nominal token : 300000

Check code : 6

Pulse ratio : 1

Token Encryption Result:

X8&DNX8X8X8&NNX

Table 3.5. Encryption Manual Calculation – Case 1

Initial Message	1	2	3	4	5	1	2	1	2	1	2	3	0	0	6
P	27	28	29	30	31	27	28	27	28	27	28	29	36	36	32
k1	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
k2	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
t = (p*k1)	270	280	290	300	310	270	280	270	280	270	280	290	360	360	320
chi = (t+k2)	274	284	294	304	314	274	284	274	284	274	284	294	364	364	324
(chi)mod 50	24	34	44	4	14	24	34	24	34	24	34	44	14	14	24
Ciphertext	X	8	&	D	N	X	8	X	8	X	8	&	N	N	X

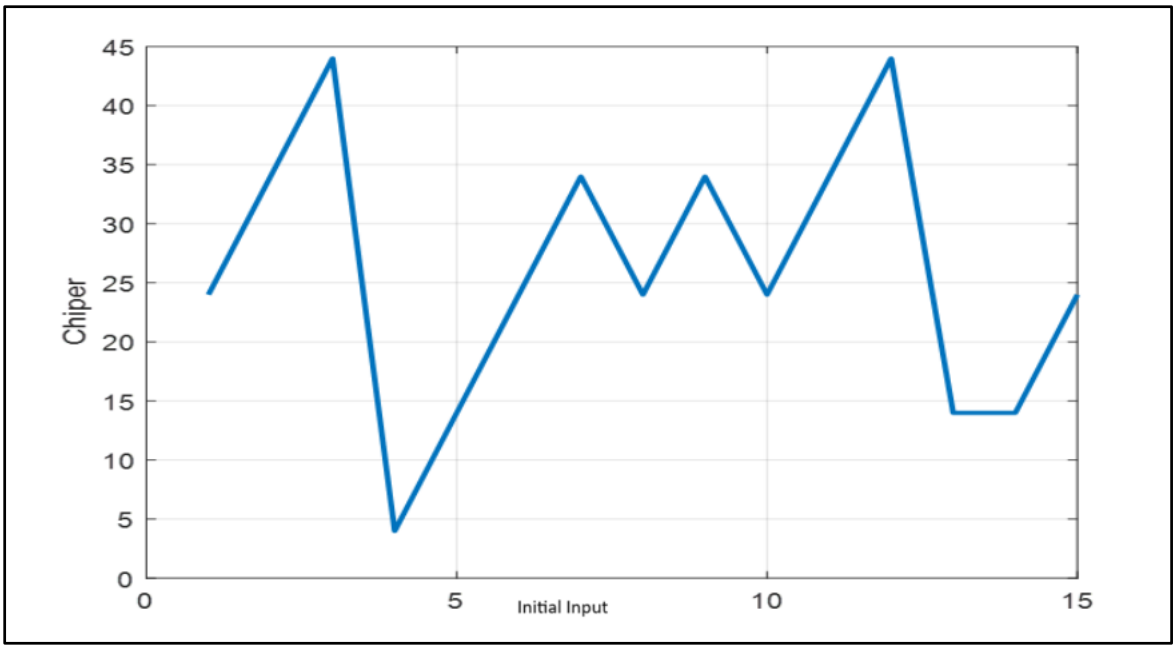


Fig.3.3. Graphic Simulation on Encryption Case#1

2.

Initial message : 800021204211001

First key : 456

Second key : 721

Nominal token : 150000

Check code : 1

Pulse ratio : 1

Token Encryption Result:

Y ,7, A,77 7

Table 3.6. Encryption Manual Calculation – Case 2

Initial Input	8	0	0	0	2	1	2	0	4	2	1	1	0	0	1
P	34	36	36	36	28	27	28	36	30	28	27	27	36	36	27
k1	456	456	456	456	456	456	456	456	456	456	456	456	456	456	456
k2	721	721	721	721	721	721	721	721	721	721	721	721	721	721	721
t = (p*k1)	15504	16416	16416	16416	12768	12312	12768	16416	13680	12768	12312	12312	16416	16416	12312
chi = (t+k2)	16225	17137	17137	17137	13489	13033	13489	17137	14401	13489	13033	13033	17137	17137	13033
(chi)mod 50	25	37	37	37	39	33	39	37	1	39	33	33	37	37	33
Ciphertext	Y	space	space	space	,	7	,	space	A	,	7	7	space	space	7

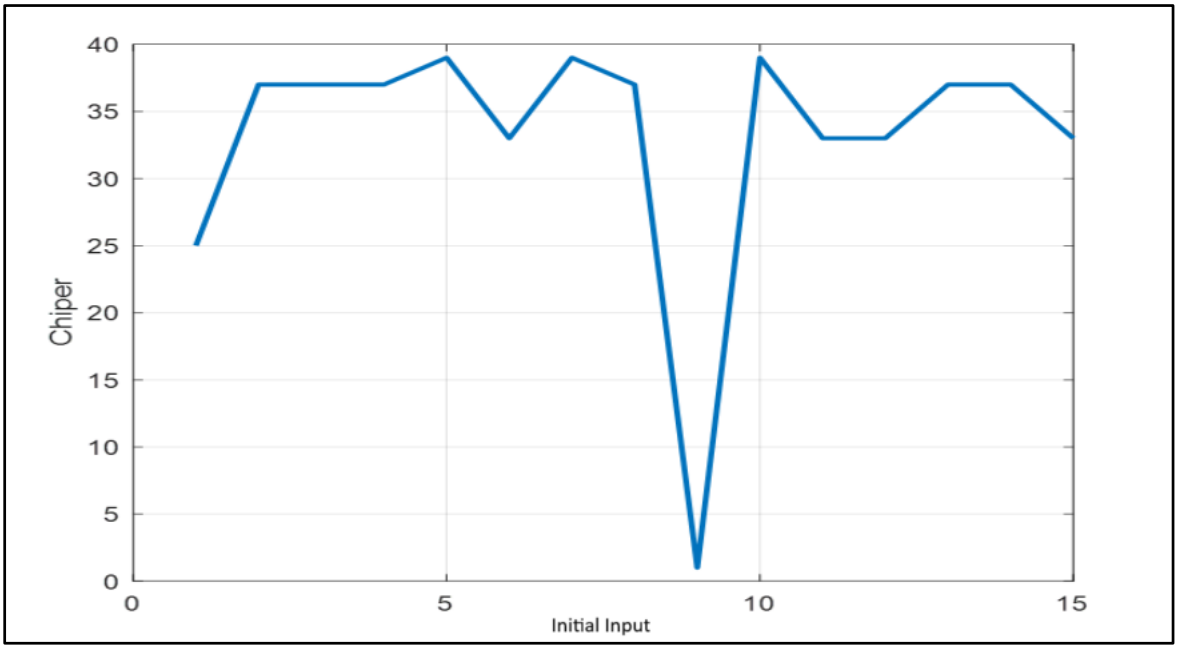


Fig.3.4. Graphic Simulation on Encryption Case#2

Simulation cases Decryption Process Vigenere Affine

Case#1

Initial message : X8&DNX8X8X8&NNX

First key : 10

Second key : 4

Nominal token : 300000

Check code : 6

Pulse ratio : 1

Token Encryption Result:

123451212123006

Proof and Calculation Table

Encryption Process

Results of table 3.6 are used as a comparison to the calculation results using the program that has been created which refer to conversion table 3.4. The code entered in the program is the original message that will be processed for encryption.

Ciphertext = X8&DNX8X8X8&NNX

1st key : 10

2nd key : 4

After the encrypted message and symmetric key are entered into the decryption program, the following values will be obtained:

Initial input =

27 28 29 30 31 27 28 27 28 27 28 29 36 36 32

Cipher =

24 34 44 4 14 24 34 24 34 24 34 44 14 14 24

Ciphertext = X8&DNX8X8X8&NNX

Vigenere transformation and *Affine cipher* will get the formula:

$$C = ((P*k1) + k2) \pmod{50} \quad (3.8)$$

T is (P*k1) with multiplier factor

$$C = (T + k2) \pmod{50} \quad (3.9)$$

- The *plaintext* sequence '1' is at number 27;

T + k2 = 274 (mod 50)

C = *24* (*ciphertext* is **'X'**)

- The *plaintext* sequence '2' is at number 28;

T + k2 = 284 (mod 50)

C = 34(*ciphertext* is **'8'**)

- The *plaintext* sequence '3' is at number 29;

T + k2 = 294 (mod 50)

C = 44(*ciphertext* is **'&'**)

- The *plaintext* sequence '4' is at number 30

T + k2 = 304 (mod 50)

C = 4(*ciphertext* is **'D'**)

- The *plaintext* sequence '5' is at number 31

T + k2 = 314 (mod 50)

C = 14(*ciphertext* is **'N'**)

- The *plaintext* sequence '1' is at number 27

T + k2 = 274 (mod 50)

C = 24(*ciphertext* nya adalah **'X'**)

- The *plaintext* sequence '2' is at number 28

T + k2 = 284 (mod 50)

C = 34(*ciphertext* is **'8'**)

- The *plaintext* sequence '1' is at number 27

T + k2 = 274 (mod 50)

C = 24(*ciphertext* is **'X'**)

- The *plaintext* sequence '2' is at number 28

T + k2 = 284 (mod 50)

C = 34(*ciphertext* is **'8'**)

- The *plaintext* '1' is at number 27

T + k2 = 274 (mod 50)

C = 24(*ciphertext* is **'X'**)

- The *plaintext sequence* '2' is at number 28

T + k2 = 284 (mod 50)

C = 34(*ciphertext* is **'8'**)

- The *plaintext* '3' is at number 29

T + k2 = 294 (mod 50)

C = 44(*ciphertext* is **'&'**)

- The *plaintext sequence* '0' is at number 36

T + k2 = 364 (mod 50)

C = 14(*ciphertext* is **'N'**)

- The *plaintext sequence* '0' is at number 36

T + k2 = 364 (mod 50)

C = 14(*ciphertext* is **'N'**)

- The *plaintext* '6' is at number 32

T + k2 = 324 (mod 50)

C = 24(*ciphertext* is **'X'**)

Decryption Process

The results of Table 3.7 are used as a comparison to the calculation results using the program that has been created refer to the conversion table 3.4. The code entered in the program is a ciphertext message that will be processed for decryption.

Table 3.7. Encryption Manual Calculation – Case 2

Ciphertext	Z	Z	0	0	/	Z	/	P	0	0	/	/	P	P	Z
(chi)mod 50	26	26	36	36	46	26	46	16	36	36	46	46	16	16	26
k1	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
k2	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
t = (chi-k2)	20	20	30	30	40	20	40	10	30	30	40	40	10	10	20
pl=t/k1	2	2	3	3	4	2	4	1	3	3	4	4	1	1	2
pl-1	1	1	2	2	3	1	3	0	2	2	3	3	0	0	1
Pesan	1	1	2	2	3	1	3	0	2	2	3	3	0	0	1

Ciphertext = ZZ00/Z/P00//PPZ

1st key = 10

2nd key = 6

After the original message and the symmetric key are entered into the encryption program, the following values will be obtained:

Initial input = s

2 2 3 3 4 2 4 1 3 3 4 4 1 1 2

Cipher =

26 26 36 36 46 26 46 16 36 36 46 46 16 16 26

Plaintext = 112231302233001

With the *Vigenere* transformation and *Affine cipher*, the formula will be obtained:

C = ((P*k1) + k2) (mod 50) (3.10)

T is (P*k1) with multiplier factor

C = (T + k2) (mod 50) (3.11)

T = (C - k2) mod 50

So will be get plaintext equation as decryption result:

P = ((C-k2)/k1) mod 50

= (t/k1) mod 50

- The *ciphertext* sequence 'Z' is at number 26;

t/k1 = 2 (mod 50)

P = *2*,

Nilai *plaintext pl* =

pl = P – 1 = (**1)**

- *The ciphertext* 'Z' is at number 26;

t/k1 = 2 (mod 50)

P = *2*

Value plaintext pl =

pl = P – 1 = (**1)**

- The ciphertext sequence '0' is at number 36;

t/k1 = 3 (mod 50)

P = *3*

Value plaintext pl =

pl = P – 1 = (**2)**

- The ciphertext sequence ‘0’ is at number 36;

t/k1 = 3 (mod 50)

P = *3*

Value plaintext pl =

pl = P – 1 = (**2)**

- The ciphertext ‘/’ is at number 46;

t/k1 = 4 (mod 50)

P = *4*

Value plaintext pl =

pl = P – 1 = (**3)**

- The ciphertext ‘z’ is at number 26;

t/k1 = 2 (mod 50)

P = *2*

Value *plaintext pl* =

pl = P – 1 = (**1)**

- The ciphertext ‘/’ is at number 46;

t/k1 = 4 (mod 50)

P = *4*

Value *plaintext pl* =

pl = P – 1 = (**3)**

- The ciphertext sequence ‘P’ is at number 16;

t/k1 = 1 (mod 50)

P = *1*

Value *plaintext pl* =

pl = P – 1 = **(0)**

- The ciphertext sequence ‘0’ is at number 36;

t/k1 = 3 (mod 50)

P = *3*

Value plaintext pl =

pl = P – 1 = **(2)**

- The ciphertext sequence‘0’ is at number 36;

t/k1 = 3 (mod 50)

P = *3*

Value plaintext pl =

pl = P – 1 = **(2)**

- The ciphertext sequence ‘/’ is at number 46;

t/k1 = 4 (mod 50)

P = *4*

Value plaintext pl =

pl = P – 1 = (**3)**

- The ciphertext sequence ‘/’ is at number 46;

t/k1 = 4 (mod 50)

P = *4*

Value plaintext pl =

pl = P – 1 = (**3)**

- The ciphertext sequence ‘P’ is at number 16;

t/k1 = 1 (mod 50)

P = *1*

Value plaintext pl =

pl = P – 1 = **(0)**

- The ciphertext sequence ‘P’ is at number 16;

t/k1 = 1 (mod 50)

P = *1*

Value plaintext pl =

pl = P – 1 = **(0)**

- The ciphertext sequence ‘Z’ is at number 26;

t/k1 = 2 (mod 50)

P = *2*,

Value *plaintext pl* =

pl = P – 1 = (**1)**

4. CONCLUSION

The *Vigenere cipher* method has a weakness due to the use of symmetric keys that are repeated so that the *ciphertext* is easily cracked. To complicate the decryption of the *ciphertext* by irresponsible parties, the modulo system is augmented with special characters by adding spaces, semicolons, periods, as well as some punctuation marks, numbers and multiplier parameters to the decryption encryption equation by applying the *Affine c*ipher formula so that it becomes a new polyalphabetic cryptography technique. This technique has a high level of diffusion and confusion properties by hiding the relationship between ciphertext and plaintext so that it will complicate cryptanalysis efforts.

5. REFERENCES

[1] Erialuode A. Henry, Omoavowere. Joy (2018): Application Platform and Token Generation Software for Prepayment Meter Administration in Electricity Distribution Companies, Machine Learning Research 3(1):1-10.

[2] Lao Ren: The Never-Ending Token Story, CLouGlobal News Metering Blog, Published on Aug 5th, 2021.

[3] Nurhayata, I G: The Development of Pre-Paid Water Meters based on AT89S52 Microcontroller, J. Physics. 2021.

[4] V. Esther Jyothi, Dr. BDCN Prasad, Dr. Ramesh Kumar Mojjada: Analysis Cryptography Encryption for Network, IOP Conference Series: Material Sciences and Engineering, Volume 981, International Conference on Recent Advancements in Engineering and Management (ICRAEM-2020) 9-10 October 2020.

[5] Yahia Alemami, Mohamad Afendee Mohamed, Saleh Atiewi: Research on Various Cryptography Techniques, International Journal of Recent technology and Engineering(IJRTE) ISSN:2277-3878, Volume-8, Issue-2S3, July 2019.

[6] Oleksii Konashevych: General Concept of Real Estate Tokenization on Blockchain, Journal European Property Law2020 -0003.