Journal of Theoretical and Applied Information Technology
15th April 2022. Vol.100. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

HYBRID CRYPTOSYSTEM USING ELGAMAL ALGORITHM AND BEAUFORT CIPHER ALGORITHM FOR DATA SECURITY

[1]HANDRIZAL, [2]FAUZAN NURAHMADI, [3]SILVIA DEWI SIREGAR
[1,2,3]Department of Computer Science, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Jl. University No. 9-A, Medan 20155, Indonesia

E-mail: handrizal@usu.ac.id

ABSTRACT

Exchanging information is very important in our public. We often exchange information in the form of files with the *.docx extension. To maintain the security of the information, it is necessary to have a data security system. In this study, a data security solution for string data is given by using cryptography. The Hybrid Cryptosystem method will be applied to the ElGamal Algorithm and the Beaufort Cipher Algorithm to provide a high level of information security. Beaufort Cipher Algorithm will be used to encrypt and decrypt files with *.docx extension and ElGamal Algorithm will be used for Beaufort Cipher's key encryption and decryption process. The results of this study indicate that the processing time required encryption and decryption process in the Beaufort Cipher Algorithm and ElGamal Algorithm is linearly compared to the length of the character in the file. The times required for the encryption process in both algorithms are longer than the time required for the decryption process.

Keywords: *Cryptography, Hybrid Cryptosystem, ElGamal, Beaufort Cipher.*

1. INTRODUCTION

Secret information security is very important since it cannot be leaked to the public [1]. The leakage of such information can cause harm to both the sender and the recipient of the information. The development of technology has brought us to the era of exchanging information via the internet. However, the internet is not a secure medium for exchanging information [2]. An outsider can easily find out the contents of information sent in the wrong way. The sender of the information needs to find a secure way to send the message so outsiders cannot obtain it. This security will be obtained by implementing the steps of encoding information in the form of a code that has a certain level of difficulty, but, that alone cannot ensure that outsiders will not be able to access the content of the information. Therefore, knowledge that focuses on the level of security of the message content is needed. One of the various sciences that understand the information security system is cryptography [3].

Cryptography is a science that gives us the possibility to maintain the security of information or message that will be sent from one place to another [4]. The cryptography process is divided into two, they are called encryption and decryption. Encryption is the process of encoding plaintext into ciphertext. While decryption is the process of returning the ciphertext into plaintext. Cryptography has various types of algorithms which could help secure data. The security of a cryptographic algorithm is created by the very important role of a key in the encryption and decryption process. A randomly determined key number will produce a random ciphertext as well. It will increase the complexity of the ciphertext and make outsiders find it more difficult to solve the algorithm used [5].

In cryptography, there are 2 types of keys used for encryption and decryption. Depending on the type of key used in each algorithm, cryptography is divided into two. The first one is symmetric cryptography. Symmetric cryptography uses a symmetric key, which means the same key will be used for both the encryption and decryption processes. This key is chosen at random and has the same size as the plaintext. The second one is asymmetric cryptography. Asymmetric cryptography is a type of cryptography that uses different keys in each encryption and decryption process. The key used in the encryption process is a public key that can be published. While the key used for the decryption process is a private key that needs to be kept as a secret by the recipient.

ElGamal is an asymmetric public key cryptosystem created in 1985 by Taher ElGamal. It is used to perform encryption and digital signature

process [6]. ElGamal algorithm is an asymmetric algorithm that has two types of keys consisting of a public key and a private key. ElGamal uses mathematical analysis in its encryption which is based on discrete logarithm problems. This algorithm consists of three processes, named the key generating process, the encryption process, and the decryption process. This algorithm is a block cipher, which performs the encryption process on plaintext blocks and produces ciphertext blocks which are then decrypted, and the results are combined back into a complete and understandable message. The advantage of the ElGamal algorithm lies in the complexity of discrete logarithms which makes the security level very good but has the disadvantage of producing ciphertext which is several times longer than plaintext.

The Beaufort Cipher is the development result of Vigenere Cipher, which is named after its inventor, Admiral Sir Francis Beaufort [7]. It is categorized as an asymmetric algorithm in cryptography. Beaufort Cipher uses a substitution encryption technique that encodes a message by using the Beaufort table and predefined keywords. An artificial table will be used to encode messages based on a key that is determined at random or by the user of this algorithm himself [8].

Hybrid Cryptosystem is a cryptographic method that combines symmetric algorithms and asymmetric algorithms to take advantage of each algorithm's strength [9]. Creating Hybrid Cryptosystem can be done by applying 2 separate cryptographic algorithms, they are a cryptographic algorithm that has an asymmetric key encapsulation scheme and a cryptographic algorithm that can as an asymmetric key encapsulation scheme [10].

In this cryptographic system, the sender generates a symmetric algorithm key. The message encryption process (plaintext) is carried out by utiliziasyan metric cryptographic algorithm that produces ciphertext, then the key from the symmetrical algorithm (plain key) is encrypted by utilizing the asymmetric cryptography algorithm's public key given by the recipient so that the cipher key is obtained. The sender will send the ciphertext and cipher key to the recipient. The recipient will decrypt the cipher key by using the asymmetric algorithm's private key to get the symmetric algorithm key (plain key). Then, this symmetric algorithm key is used to decrypt the ciphertext. Thus, the recipient gets the plaintext.

2. METHOD

There will be three steps to be done for this system. They are key generator, encryption, and decryption. The key generator will generate the public and private Encrypting consisting of encrypting plaintext into ciphertext using the Beaufort key and encrypting the Beaufort key into cipher key using the ElGamal public key. And decrypting consists of returning the cipher key into the original Beaufort key using the ElGamal private key and ciphertext into plaintext using the Beaufort key.

2.1. Key Generator

In the generating key process, it is necessary to have a prime number p, a primitive element g, and an arbitrary number x, using the terms that the value of g and x is less than p. The public key for encryption is generated from these 3 numbers with the following equation.

$$y = g^x \bmod p$$

Steps that need to be done to get ElGamal's public key and private key:

1. Choose a prime number p at random, provided that $p > 255$.
2. To see whether p is a prime number or not, use the Agrawal Biswas test.
3. Generate a random Z, provided that $2 \leq Z < p - 2$ and GCD $(p, Z) = 1$.
4. If $(1 + Z)^p \ (mod\ p) \equiv 1 + Z^p \ (mod\ p)$, then p is a prime number.
5. Generate a random primitive root of p called g, provided that $g < p$.
6. Generate a random number x, provided that $1 < x < p$-2.
7. Compute $y=g^x \ mod\ p$.
8. Keep the public key (p, g, y) and private key (p, x).

The recipient of the message is in charge of generating a private key and a public key and then sending the public key to the sender of the message. The sender does not need the private key because the key is only needed for the decryption process.

2.2. Encryption

That is a process in which a text message (plaintext) is encoded into a message that has confidentiality or ciphertext. This process requires public keys and the private keys process consists of plaintext encryption and the Beaufort key encryption.

2.2.1. Plaintext Encryption

In the Beaufort Cipher table, the characters in the first row represent the plaintext, while the characters in the first column represent the ciphertext. The encryption process is carried out by drawing a line from one

plaintext character vertically downwards until it finds a predetermined key character, then pulling a horizontal line to the left until the first column in the same row. From this process obtained ciphertext. The decryption process is carried out by drawing a line horizontally to the right starting from the ciphertext character from the first column as the starting point until it finds the key character, then with the key character as the starting point, draw a vertical line up to the character in the first line. From this process obtained plaintext.

Plaintext will be encrypted using the Beaufort key. The steps are:

1. Choose a plaintext
2. Convert all of the characters in plaintext to ASCII values.
3. Generate a random number k as the key for each character of the plaintext, provided that $0 \leq k < 256$.
4. Encrypt the plaintext by computing $Ci \equiv Ki - Pi \ (mod\ 256)$. Pi is the character of plaintext, $Kthe$he s key, and the Ci is ciphertext.
5. Convert all of the results to char.
6. Save the ciphertext.

2.2.2. Beaufort Key Encryption

That is a process where the Beaufort key (plain key) is encoded into a key that has confidentiality or a cipher key. This process requires the public keys p, g, and y.

Beaufort key will be encrypted using ElGamal's public key. The steps are:

1. Choose the Beaufort key.
2. Divide the characters of Beaufort key into blocks m1, m2, m3, ..., mn with block m values the in top range *0* top-*1*.
3. Convert each block to ASCII values.
4. Generate a random number k as much as the number of m, provided that $1 \leq k \leq p-1$.
5. Encrypt the Beaufort key using ElGamal's public key (p, g, y) by computing $a = g^k \ mod\ p$ and $b = YK.\ m\ mod\ p$.
6. Arrange the encrypted blocks sequentially from a1, b1, a2, b2, …, an, bn to get the cipher key.

2.3. Decryption

Decryption will be divided into cipher key decryption and ciphertext decryption.

2.3.1. Cipher Key Decryption

The decryption process is the process of returning a message that has previously been encrypted (ciphertext) into the original message or plaintext. This process requires the private keys p and x to decrypt a and b with the following:

$$m = b\ .\ a^{p-1-x} \bmod p$$

The cipher key will be returned to be the original Beaufort key by using ElGamal's private key. The steps are:

1. Choose the cipher key.
2. Get the private key (p, x).
3. Compute $m = b.\ a^{p-1-x}\ mod\ p$.
4. Convert m values to char based on Athe SCII table.
5. Arrange all chars in the order m1, m2, m3, ..., mn to get the Beaufort key.

2.3.2. Ciphertext Decryption

The ciphertext will be decrypted using the Beaufort key. The steps are declared as follows:

1. Choose the ciphertext.
2. Convert all of the characters in ciphertext to ASCII values.
3. Get the Beaufort key.
4. Decrypt each character of ciphertext by computing $Pi \equiv Ki - Ci \ (mod\ 256)$
5. Convert Pi values to char based on SCIthe I table.

3. RESULTS AND DISCUSSION

System testing and analysis is carried out on encryption and decryption with processing time criteria

3.1. Generate Key

Generate the public key and private key using the ElGamal algorithm.

1. Generate a random number $p = 347$ and test it with Agrawal Biswas:
   a. Generate a random $Z = 50$ $(2 \leq Z < p - 2)$.
   b. GCD $(p, Z) = (347,50) = 1$
   c. Calculate:
   $(1 + Z)^p \ (mod\ p) = 1 + Z^p \ (mod\ p)$
   $(1 + 50)^{347}\ mod\ 347 = 1 + 50^{347}\ mod\ 347$
   $51 = 51$
2. Generate a random number $g = 7$ as primitive root of p.
3. Generate a random number $x = 13$ $(1 < x < p\text{-}2)$.
4. Compute $y = g^x\ mod\ p = 7^{13}\ mod\ 347 = 57$.
5. Keep $p = 347$, $g = 7$, $y = 57$ as public key and $x = 13$ as private key.

3.2. Encryption

3.2.1. Plaintext Encryption

Encrypt the plaintext using the Beaufort Cipher algorithm.

1. Choose a plaintext. For example, plaintext (Pi) is "STREET".
2. Convert plaintext into ASCII code.

Table 1. Plaintext to ASCII Code

Pi	
Char	Index based on ASCII
S	83
T	84
R	82
E	69
E	69
T	84

3. Generate a random key, *Ki* = "{'Z2#~" ($0 \leq K < 256$), as the Beaufort key.
4. Convert key into ASCII code.

Table 2. Beaufort Key to ASCII Code

Ki	
Char	Index based on ASCII
{	123
'	146
Z	122
2	50
#	35
~	126

5. Calculate:
   $Ci = Ki - Pi \pmod{256}$

   $P_1 = 83, K_1 = 123$
   $C_1 = 123 - 83 \pmod{256}$
   $C_1 = 40$

   $P_2 = 84, K_2 = 146$
   $C_2 = 146 - 84 \pmod{256}$
   $C_2 = 62$

   $P_3 = 82, K_3 = 122$
   $C_3 = 122 - 82 \pmod{256}$
   $C_3 = 40$

   $P_4 = 69, K_4 = 50$
   $C_4 = 50 - 69 \pmod{256}$
   $C_4 = 237$

   $P_5 = 69, K_5 = 35$
   $C_5 = 35 - 69 \pmod{256}$
   $C_5 = 222$

   $P_6 = 84, K_6 = 126$
   $C_6 = 126 - 84 \pmod{256}$
   $C_6 = 42$
6. Convert *Ci* to ciphertext.

Table 3. Ci to Ciphertext Using the ASCII Code

Ci	
Index based on ASCII	Char
40	(
62	>
40	(
237	í
222	Þ
42	*

7. Then, ciphertext = "(>(í Þ*".

3.2.2. Beaufort Key Encryption

1. Get the Beaufort key (*K* = "{'Z2#~").
2. Beaufort key will be considered as the plain message (plain key) in the ElGamal algorithm process. So, *m* will be:

Table 4. mi to ASCII

mi	
Char	Index based on ASCII
{	123
'	146
Z	122
2	50
#	35
~	126

3. Get the ElGamal public key ($p = 347$, $g = 7$, $y = 57$).
4. Generate a random k for each char in plain key ($1 \leq k \leq p\text{-}1$).

Table 5. Random ki for mi

m_i	k_i
123	11
146	7
122	5
50	3
35	8
126	13

5. Calculate $a = g^k \bmod p$ and $b = y^k . m \bmod p$.
   $m_1 = 123, k_1 = 11$
   $a_1 = g^k \bmod p$
   $a_1 = 7^{11} \bmod 347$

$a_1 = 334$

$b_1 = y^k . m_1 \ mod \ p$
$b_1 = 57^{11} . 123 \ mod \ 347$
$b_1 = 44$

$m_2 = 146, k_2 = 7$
$a_2 = g^k \ mod \ p$
$a_2 = 7^7 \ mod \ 347$
$a_2 = 112$

$b_2 = y^k . m_1 \ mod \ p$
$b_2 = 57^7 . 146 \ mod \ 347$
$b_2 = 127$

$m_3 = 122, k_3 = 5$
$a_3 = g^k \ mod \ p$
$a_3 = 7^5 \ mod \ 347$
$a_3 = 151$

$b_3 = y^k . m_3 \ mod \ p$
$b_3 = 57^5 . 122 \ mod \ 347$
$b_3 = 184$

$m_4 = 50, k_4 = 3$
$a_4 = g^k \ mod \ p$
$a_4 = 7^3 \ mod \ 347$
$a_4 = 343$

$b_4 = y^k . m_4 \ mod \ p$
$b_4 = 57^3 . 50 \ mod \ 347$
$b_4 = 302$

$m_5 = 35, k_5 = 8$
$a_5 = g^k \ mod \ p$
$a_5 = 7^8 \ mod \ 347$
$a_5 = 90$

$b_5 = y^k . m_5 \ mod \ p$
$b_5 = 57^8 . 35 \ mod \ 347$
$b_5 = 345$

$m_6 = 126, k_6 = 13$
$a_6 = g^k \ mod \ p$
$a_6 = 7^{13} \ mod \ 347$
$a_6 = 57$

$b_6 = y^k . m_6 \ mod \ p$
$b_6 = 57^{13} . 126 \ mod \ 347$
$b_6 = 178$

Table 6. Cipher block (ai, bi)

m_i	a_i	b_i
123	334	44
146	112	127
122	151	184
50	343	302
35	90	345
126	57	178

6. Keep the cipher key (a_i, b_i).

3.3. Decryption

3.3.1. Cipher Key Decryption

Cipher key will be returned to be the original Beaufort key by using ElGamal's private key with the following steps:

1. Get the cipher key (a_i, b_i).
2. Get the private key (p, x) = (347, 13).
3. Calculate:
   $m_i = b_i . AIP^{-1-x} \ mod \ p$
   $a_1 = 334, b_1 = 44$
   $m_1 = 44 . 334^{333} \ mod \ 347$
   $m_1 = 123$

   $a_2 = 112, b_2 = 127$
   $m_2 = 127 . 112^{333} \ mod \ 347$
   $m_2 = 146$

   $a_3 = 151, b_3 = 184$
   $m_3 = 184 . 151^{333} \ mod \ 347$
   $m_3 = 122$

   $a_4 = 343, b_4 = 302$
   $m_4 = 302 . 343^{333} \ mod \ 347$
   $m_4 = 50$

   $a_5 = 90, b_5 = 345$
   $m_5 = 345 . 90^{333} \ mod \ 347$
   $m_5 = 35$

   $a_6 = 57, b_6 = 178$
   $m_6 = 178 . 57^{333} \ mod \ 347$
   $m_6 = 126$

4. Return m_i to plainkey.

Table 7. mi to Plainkey

mi	
Index based on ASCII	Char
123	{
146	'
122	Z
50	2
35	#
126	~

5. So, plain key (m_i) = "{'Z2#~".

3.3.2. Ciphertext Decryption

The ciphertext will be decrypted using the Beaufort key with the following steps:

1. Get the ciphertext = "(>(í Þ*" and convert it to ASCII code.

Table 8. Cipherkey to ASCII Code

Ci	
Char	Index based on ASCII
(	40
>	62
(	40
í	237
Þ	222
*	42

2. Beaufort key (K) = "{'Z2#~".
3. Convert Beaufort Key to ASCII code.

Table 9. Beaufort Key to ASCII Code

Ki	
Char	Index based on ASCII
{	123
'	146
Z	122
2	50
#	35
~	126

4. Calculate:
   $Pi = Ki - Ci \ (mod\ 256)$

   $C_1 = 40,\ K_1 = 123$
   $P_1 = 123 - 40 \ (mod\ 256)$
   $P_1 = 83$

   $C_2 = 62,\ K_2 = 146$
   $P_2 = 146 - 62 \ (mod\ 256)$
   $P_2 = 84$

   $C_3 = 40,\ K_3 = 122$
   $P_3 = 122 - 40 \ (mod\ 256)$
   $P_3 = 82$

   $C_4 = 237,\ K_4 = 50$
   $P_4 = 50 - 237 \ (mod\ 256)$
   $P_4 = 69$

   $C_5 = 222,\ K_5 = 35$
   $P_5 = 35 - 222 \ (mod\ 256)$
   $P_5 = 69$

   $C_6 = 42,\ K_6 = 126$
   $P_6 = 126 - 42 \ (mod\ 256)$
   $P_6 = 84$

5. Convert P_1 to plaintext.

Table 10. Pi to Plaintext

Pi	
Index based on ASCII	Char
83	S
84	T
82	R
69	E
69	E
84	T

That way, we get back the same plaintext as the original plaintext.

3.4. Real Running Time

3.4.1. Encryption Time

The time for the encryption process is the time it takes to complete the encryption process on the plaintext file by the Beaufort key and the Beaufort key encryption by the ElGamal public key. The length of the plaintext is the same as the length of the Beaufort key.

In this testing process, 6 texts are used consisting of small, medium, and large files. Small files have less than 100 characters, medium files have less than 5000 characters, and large files have 10000 characters or more. The test is carried out three times for each text and the average processing time will be calculated in milliseconds.

Table 11. Encryption Process Time

Text length (plaintext in char)	Processing Time (millisecond) 1st Test	2nd Test	3rd Test	4th Test	5th Test	Average
10	10	16	6	3	3	7,6
50	19	14	13	11	19	15,2
100	106	24	41	24	44	47,8
150	86	61	68	47	91	70,6
1000	305	369	232	213	220	267,8
10000	2176	4929	2921	2971	2654	3130,2

Table 12. Decryption Process Time

Text length (plaintext in char)	Processing Time (millisecond) 1st Test	2nd Test	3rd Test	4th Test	5th Test	Average
10	6	9	2	2	2	4,2
50	19	7	4	4	3	7,4
100	40	15	24	14	18	22,2
150	44	45	49	34	58	46
1000	103	127	145	142	77	118,8
10000	770	1797	418	1090	613	937,6

Table 11 shows that each testing time is different for the encryption process with several plaintexts that have different character lengths.

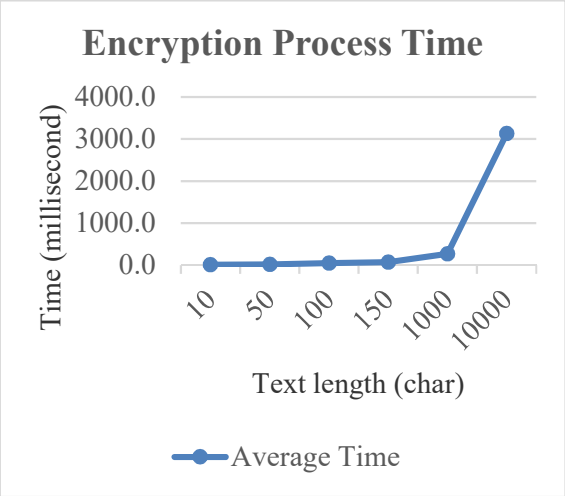

Figure 1. The Result of Encryption

Figure 1 shows that the length of time required for the encryption process is influenced by the length of the plaintext character. The longer the plaintext character, the longer it will take to complete the encryption process.

3.4.2. Decryption Time

The time for the decryption process is the time required for the system to convert the cipher key into a Beaufort key with ElGamal's private key and convert the ciphertext into plaintext with the Beaufort Cipher key.

In the testing process, the system used 6 ciphertexts with a length of 10, 50, 100, 150, 1000, and 10000 characters which are the results of the previous encryption test. Each test is carried out five times and the average processing time will be calculated in milliseconds for each ciphertext length.

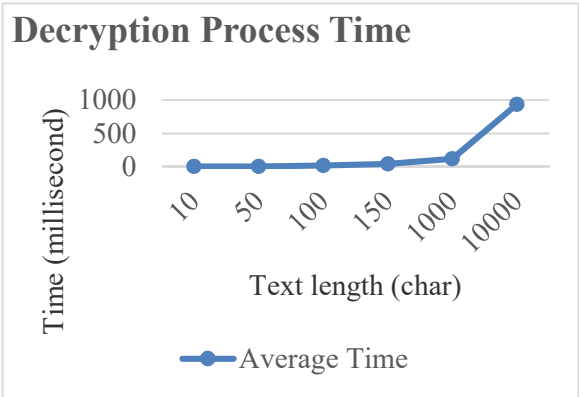

Figure 2. The Result of Decryption

Figure 2 shows that the length of time required for the decryption process is influenced by the length of the ciphertext character used. The longer the ciphertext character, the longer the time to complete the decryption process.

Based on Figure 1 and Figure 2 it can be seen that the time required for the encryption and decryption process using the hybrid cryptosystem method is directly proportional to the length of the plaintext and ciphertext. Which means, the longer the character used, the longer it takes for the encryption and decryption process. The hybrid cryptosystem method using the Elgamal algorithm and Beaufort cipher algorithm can cover the weakness of the ElGamal algorithm in terms of faster encryption and decryption process time.

4. CONCLUSION

The conclusion obtained is that the time required for the encryption process is longer when compared to the decryption process. This is because the encryption process in the ElGamal Algorithm causes the text size to increase to twice the previous text size. The hybrid cryptosystem method using the Elgamal algorithm and Beaufort cipher algorithm can cover the weakness of the ElGamal algorithm in terms of

faster encryption and decryption process time. The encryption process by using the Beaufort Cipher produces a cipher key that has the same length as the plaintext length and the decryption process by using ElGamal produces the same plaintext as the original plaintext. The time required for the encryption and decryption process is directly proportional to the length of the text used.. Hybrid cryptosystem method using Elgamal algorithm and Beaufort cipher algorithm can cover the weakness of the ElGamal algorithm in terms of faster encryption and decryption process time. The application of the hybrid cryptosystem method for key generation, encryption and decryption for data security was successfully carried out by combining the ElGamal Algorithm and the Beaufort Cipher Algorithm.

REFERENCES

[1]. Hadi, A. S. 2016. Information hiding in Linked Opened Data. Journal of the University of Babylon, 24(3).

[2]. Hidajat, M. S., & Setiarso, I. 2019. Securing Digital Color Image based on Hybrid Substitution Cipher. Journal of Applied Intelligent System, 4(2), 86-95.

[3]. Okeyinka, A. E. 2015. Computational complexity study of RSA and Elgamal algorithms. In The World Congress on Engineering and Computer Science (pp. 233-243). Springer, Singapore.

[4]. Chowdhary, C. L. et all. 2020. Analytical study of hybrid techniques for image encryption and decryption. *Sensors*, *20*(18), 5162.

[5]. Pai, D. G., & Pai, Y. 2021. Analysis of the Beaufort Cipher Expansion Technique and Its Usage in Providing Data Security in Cloud. In Cyber Intelligence and Information Retrieval (pp. 49-58). Springer, Singapore.

[6]. Rachmawati, D. & Sharif, A. 2019. Hybrid Cryptosystem Combination Algorithm of Hill Cipher 3x3 and Elgamal To Secure Instant Messaging for Android. Journal of Physics: Conference Series (Vol. 1235, No. 1, p. 012074). IOP Publishing.

[7]. Saraswat, A., Khatri, C., Thakral, P., & Biswas, P. 2016. An extended hybridization of Vigenere and Caesar cipher techniques for secure communication. *Procedia Computer Science*, *92*, 355-360.

[8]. Sari, R. N. & Hayati, R. S. 2018. Beaufort Cipher Algorithm Analysis Based on the Power Lock-Blum Blum Shub in Securing Data. In 2018 6th International Conference on Cyber and IT Service Management (CITSM) (pp. 1-4). IEEE.

[9]. Agrawal, A., & Patankar, G. 2016. Design of hybrid cryptography algorithm for secure communication. International Research Journal of Engineering and Technology (IRJET), 3(01), 2395-0056.

[10]. Owolabi, O. Y., Shola, P. B., & Jibrin, M. B. 2017. Improved Data Security System Using Hybrid Cryptosystem. 201.7 IJSRSET, 3(3).