DATA SCIENCE
Journal of Computing and Applied Informatics

Contents lists available online at TALENTA Publisher

DATA SCIENCE: JOURNAL OF COMPUTING AND APPLIED INFORMATICS (JoCAI)

Journal homepage: https://jocai.usu.ac.id

Performance Analysis of Hybrid Cryptographic Algorithms Rabbit Stream and Enhanced Dual RSA

Demonius Sarumaha[1], Mohammad Andri Budiman[2], and Muhammad Zarlis[3]

[1,2]*Master of Informatics Program, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia*

[3]*Department of Information System, Bina Nusantara University, Jakarta, Indonesia*

ARTICLE INFO

Article history:
Received 12 December 2022
Revised 26 January 2023
Accepted 27 January 2023
Published online 30 January 2023

Keywords:
Rabbit Stream
Enhanced Dual RSA
Cryptography
Hybrid

Email:
[1]demonius213@gmail.com
[2]mandrib@usu.ac.id
[3]muhammad.zarlis@binus.edu

Corresponding Author:
Mohammad Andri Budiman

ABSTRACT

Cryptography is a technique for encoding data by encrypting plaintext into an unreadable (meaningless) form. Cryptographic methods have good and bad performance depending on the type of algorithm we use. Therefore, the purpose of this study is to measure speed by combining the two algorithms used. The Rabbit Stream algorithm is a stream cipher algorithm whose system security depends on the generation of a key bit stream (keystream), which only guarantees 128-bit key security but has the advantage of being fast in the encryption and decryption process, while the Enhanced Dual RSA algorithm is an asymmetric algorithm to increase data protection from the Dual RSA algorithm by utilizing the Pells equation as a substitute for public key exponents. On the other hand, the algorithm in question requires a significant amount of time to encrypt messages with a large capacity when compared to the Rabbit Stream algorithm. Nonetheless, the study's findings suggest that using a hybrid method is comparatively faster for processing substantial amounts of data.

IEEE style in citing this article:
D. Saruhama, M.A.Budiman and M. Zarlis, " Performance Analysis of Hybrid Cryptographic Algorithms Rabbit Stream and Enhanced Dual RSA", *Data Science: Journal of Computing and Applied Informatics (JoCAI)*, vol. 7, no. 1, pp. 35-43, 2023.

1. Introduction

The development of technology in this era is a very important thing. Technology is able to help various organizations, companies, or other parties in communicating and exchanging data or documents. Along with the development of the times, agencies tend to use electronic documents in exchanging data because it is very easy and fast, but in data exchange, theft is often committed by third parties for personal gain [1]. Therefore data security is very important to keep confidential and to maintain the confidentiality of the data, and cryptographic techniques are needed [2].

Data security using cryptographic techniques is one way to hide the original message in another form that cannot be accessed or modified by unauthorized persons. Cryptography can be classified into two types based on the key used, namely symmetric cryptography and asymmetric cryptography. Symmetric cryptography is a cryptographic algorithm that uses the same key in the encryption and decryption processes. The communicating entities must exchange keys so that they can be used in the decryption process. The secret key used by the sender and receiver can be a series of random letters and numbers. Examples of symmetric algorithms: the Spritz algorithm, Rabbit Stream, RC4, TwoFish, Rijndael, etc [3]. Asymmetric cryptography is a public key algorithm that uses two different keys in the encryption and decryption processes, namely one for encryption and one for decryption. The public key used for encryptsion can be known publicly, while the private key is not desired [4].

In asymmetric cryptography, one of the keys is published, and knowing the public key will be an opportunity for unauthorized persons to decipher all the keys, even though it requires a long process. The advantage of using an asymmetric algorithm is that it provides greater scalability than a symmetric algorithm and guarantees confidentiality and authentication, but an asymmetric algorithm works much slower than a symmetric [5]. Using a symmetric algorithm can often be penetrated by cryptanalysts easily because the security of a symmetric algorithm depends only on the secrecy of the key. If the key used can be guessed or known by irresponsible parties, then all messages can be easily decrypted [6].

The Rabbit Stream algorithm is a stream cipher cryptographic algorithm that only guarantees the secrecy of a 128-bit key and uses the same key in encrypting and decrypting messages. The encryption and decryption process is carried out by XOR the generated key with plaintext or ciphertext [7]. The Rabbit Stream algorithm has a very fast process in key generation, encryption, and decryption processes. The security of the RSA algorithm, in general, does not only lie in the large number of key characters generated but actually lies in the difficulty of factoring very large numbers into prime numbers. The purpose of factoring is to get the public key and private key [8]. The Enhanced Dual RSA algorithm is an asymmetric algorithm to increase data protection from the Dual RSA algorithm by utilizing the Pells equation as a substitute for public key exponents [9]. The Enhanced Dual RSA cryptographic algorithm is less efficient for encrypting large messages because it produces a ciphertext that increases many times, causing problems, namely where it takes a long time during the encryption process and in sending the ciphertext. So this study combined the Enhanced Dual RSA algorithm with Rabbit Stream in a hybrid scheme to solve existing problems.

2. Methods

At this stage, to speed up the process of encrypting and decrypting large-capacity data, researchers use a hybrid method. Hybrid cryptography is a cryptographic technique that takes advantage of the advantages of each algorithm by combining symmetric and asymmetric algorithms. In hybrid cryptography, the sender generates a symmetric algorithm key, and then the plaintext will be encrypted using the private key to produce a ciphertext. The symmetric algorithm private key is encrypted using the asymmetric algorithm public key so as to get the cipherkey. Then the ciphertext and cipherkey will be sent to the recipient. To get the contents of the message sent, the recipient decrypts the cipherkey using the asymmetric private key algorithm and gets the symmetric private key. Furthermore, this symmetrical private key will be used to decrypt the ciphertext so as to get the original plaintext. The algorithms used in this study are the Rabbit Stream algorithm and the Enhanced Dual RSA algorithm.

2.1. Rabbit Stream Algorithm

The Rabbit Stream algorithm was first published publicly in 2003, Rabbit Stream is a stream cipher that only guarantees the security of a secret key of 128 bit. where the encryption and decryption processes are performed by XOR bit of plaintext with the key to be converted into ciphertext [10]. The steps in the process of encryption and decryption of the Rabbit Stream algorithm are as follows.

1. Key generation process
   a. Key set up scheme

      The first step of this algorithm is to determine and expand the key size of 128 bits which is divided into eight sub-keys: $k_0 = K^{[15\dots0]}$, $k_1 = K^{[31\dots16]}$,……, $k_7 = K^{[127\dots112]}$. The state and counter variables are initialized from the subkey as follows:

$$xj,o \begin{cases} k_{(j+1\ mod\ 8)} \,||\, k_j & \text{if } j \text{ even} \\ k_{(j+5\ mod\ 8)} \,||\, k_{(j+4\ mod\ 8)} & \text{if } j \text{ odd} \end{cases}$$

and

$$cj,o \begin{cases} k_{(j+4\ mod\ 8)} \,||\, k_{(j+5\ mod\ 8)} & \text{if } j \text{ even} \\ k_j \,||\, k_{(j+1\ mod\ 8)} & \text{if } j \text{ odd} \end{cases}$$

Modify the status counter by following these steps:

$$c_{j,4} = c_{j,4} \oplus x_{(j+4\ mod\ 8)}$$

b. Initialization Vector (IV))

In this scheme, the next step is to modify the status counter as an Initialization Vector function by XOR a 64-bit Initialization Vector. And iterate four times.

$$c0{,}4 = c0{,}4 \oplus IV^{[31\ldots0]} \qquad c1{,}4 = c0{,}4 \oplus IV^{[63\ldots48]} \,||\, IV^{[31\ldots16]}$$

$$c2{,}4 = c2{,}4 \oplus IV^{[63\ldots32]} \qquad c3{,}4 = c3{,}4 \oplus IV^{[47\ldots32]} \,||\, IV^{[15\ldots0]}$$

$$c4{,}4 = c4{,}4 \oplus IV^{[31\ldots0]} \qquad c5{,}4 = c5{,}4 \oplus IV^{[63\ldots48]} \,||\, IV^{[31\ldots16]}$$

$$c6{,}4 = c6{,}4 \oplus IV^{[63\ldots32]} \qquad c7{,}4 = c7{,}4 \oplus IV^{[47\ldots32]} \,||\, IV^{[15\ldots0]}$$

c. Counter system

Counter dynamics is defined as follows:

$$co, i+1 = co, i + \alpha 0 + \Phi 7, i \; mod \; 2^{32}$$

$$c1, i+1 = c1, i + \alpha 1 + \Phi o, i+1, mod \; 2^{32}$$

$$c2, i+1 = c2, i + \alpha 2 + \Phi 1, i+1, mod \; 2^{32}$$

$$c3, i+1 = c3, i + \alpha 3 + \Phi 2, i+1, mod \; 2^{32}$$

$$c4, i+1 = c4, i + \alpha 4 + \Phi 3, i+1, mod \; 2^{32}$$

$$c5, i+1 = c5, i + \alpha 5 + \Phi 4, i+1, mod \; 2^{32}$$

$$c6, i+1 = c6, i + \alpha 6 + \Phi 5, i+1, mod \; 2^{32}$$

$$c7, i+1 = c7, i + \alpha 7 + \Phi 6, i+1, mod \; 2^{32}$$

Where is the counter carry bit, $\Phi j, i+1$ based on

$$\Phi j, i+1 \begin{cases} 1 & if \; co, i + \alpha o + \Phi 7, i \geq 2^{32} \wedge j = 0 \\ 1 & if \; cj, i + \alpha j + \Phi j - 1, i+1 \geq 2^{32} \wedge j > 0 \\ 0 & \text{on the contrary} \end{cases}$$

Before calculating the following state function, the system counter is updated by following the steps above:

$a0 = 0x4d34d34d$

$a1 = 0xd34d34d3$

$a2 = 0x34d34d34$

$a3 = 0x4d34d34d$

$a4 = 0xd34d34d3$

$a5 = 0x34d34d34$

$a6 = 0x4d34d34d$

$a7 = 0xd34d34d3$

d. Next state function

The essence of the Rabbit Stream algorithm is the iteration of the system, which is defined by the following equation:

$$gj, i = \left(\left(xj, i + cj, i+1\right)^2 \oplus \left(\left(xj, i + cj, i+1\right)\right)^2 \gg 32\right)\right) mod \; 2^{32}$$

The next step is using the formula:

$$x_0 = G_0 + (G_7 \lll 16) + (G_6 \lll 16) \; mod \; 2^{32}$$

$$x_1 = G_1 + (G_0 \lll 8) + G_7 \; mod \; 2^{32}$$

$$x_2 = G_2 + (G_1 \lll 16) + (G_0 \lll 16) \; mod \; 2^{32}$$

$$x_3 = G_3 + (G_2 \lll 8) + G_1 \; mod \; 2^{32}$$

$$x_4 = G_4 + (G_3 \lll 16) + (G_2 \lll 16) \; mod \; 2^{32}$$

$$x_5 = G_5 + (G_4 \lll 8) + G_3 \; mod \; 2^{32}$$

$x_6 = G_6 + (G_5 \lll 16) + (G_4 \lll 16)\ mod\ 2^{32}$

$x_7 = G_7 + (G_6 \lll 8) + G_5\ mod\ 2^{32}$

e. Extraction scheme

$s_i^{(15...0)} = x_{0,i}^{(15...0)} \oplus x_{5,i}^{(31...16)}$ $s_i^{(31...16)} = x_{0,i}^{(31...16)} \oplus x_{3,i}^{(15...0)}$

$s_i^{(17...32)} = x_{2,i}^{(15...0)} \oplus x_{7,i}^{(31...16)}$ $s_i^{(63...48)} = x_{2,i}^{(31...16)} \oplus x_{5,i}^{(15...0)}$

$s_i^{(79...64)} = x_{4,i}^{(15...0)} \oplus x_{1,i}^{(31...16)}$ $s_i^{(31...16)} = x_{4,i}^{(31...16)} \oplus x_{7,i}^{(15...0)}$

$s_i^{(111...96)} = x_{6,i}^{(15...0)} \oplus x_{3,i}^{(31...16)}$ $s_i^{(127..112)} = x_{6,i}^{(31.16)} \oplus x_{1,i}^{(15..0)}$

2. Encryption and decryption process

Encryption $c_i = p_i \oplus s_i$

Decryption $p_i = c_i \oplus s_i$

The steps below will show the process of encryption and decryption of the Rabbit Stream algorithm as an example, as follows.

1. Input 128-bit Rabbit key
   Key = DEMONIUSSARUMAHA
   Plaintext = UNIVERSITAS SUMATERA UTARA
2. The results of the key generation process by utilizing the predetermined key are: "$cf4c1b2c4b8319dc880c5fab4fa58b52$"
3. The encryption process is that the key that has been obtained from the generation process is XOR with each two-digit hexadecimal number with plaintext, resulting in: "9a02527a0ed14a95dc4d0c8b1cf0c6139b09496d6bd64d9dda4d"
4. The decryption process, namely the ciphertext and key received, is XOR with two digits each and produces the original plaintext.

2.2. Enhanced Dual RSA Algorithm

The RSA algorithm is a public key algorithm invented by Ron Rivest, Adi Shamir, and Leonard Adleman in 1978. The RSA algorithm uses two keys to encrypt and decrypt messages [11]. Dual RSA is a development algorithm from RSA used to reduce memory usage and increase the security of the RSA algorithm [12]. The Enhanced Dual RSA algorithm is a public key algorithm to increase the security of the Dual RSA algorithm by using Pell's equation to hide exponential public keys and pseudo modulus to avoid factorization attacks. The proposed system uses the number z as the pseudo modulus, so it is very difficult to infer the value of n. The steps in the encryption and decryption process of the Enhanced Dual RSA algorithm are as follows.

1. Key generation process
   a. Choose four prime numbers p, q, r, s where n can be calculated by:
      $n = p.q.r.s$
   b. Compute Totient n with::
      $\Phi(n) = (p-1)*(q-1)*(r-1)*(s-1)$
   c. Choose a prime z to replace n where $\Phi(n) < z < n$,
   d. Find $\Phi(z)$ provided that::
      $\Phi(z) = z - 1$
   e. Randomly find the exponents of the public keys e and f was given the following provision:
      - odd numbers
      - $\text{GCD}(\Phi(z), e) = 1$ and $\text{GCD}(\Phi(z), f) = 1$
   f. Find the private key component d with provision:
      $(e * f * d)\ mod\ \Phi(z) = 1$
   g. Find the partner of (x1,y1) and (x2,y2) with provision: $x_1^2 - ey_1^2 = 1\ dan\ x_2^2 - fy_2^2 = 1$

h. Commonly shared keys are ($x1$, $y1$, $x2$, $y2$, z)

2. Encryption and decryption process

Encryption $C = \left(M^{(x1^2-1)/y1^2}\right) mod\ z^{\ (x2^2-1)/y2^2}\ mod\ z$
Decryption $M = C^d\ mod\ z$
The steps below will show the encryption and decryption process of the Enhanced Dual RSA algorithm an example, as follows:

1. Key generation process
   a. Choose a prime number at random$p = 89, q = 37, r = 29, s = 43$
      $n = 4106371$
   b. Calculate Totient n with:
      $\Phi(n) = 3725568$
   c. Select the prime number z to replace n where Φ(n) < z <n,$z = 3929183$
   d. Find $\Phi(z)$:
      $\Phi(z) = 3929182$
   e. Find the exponents of the *e* and *f* public keys at random
      value *e* = 89 and *f* = 43
   f. Find the private key component *d*
      d = 356265
   g. Find the partner of ($x1$,$y1$) and ($x2$,$y2$)
      $x1 = 500001,\ y1 = 53000$ and $x2 = 3482,\ y2 = 531$
   h. Commonly shared keys are ($x1$, $y1$, $x2$, $y2$, z)

2. Encryption and decryption process

Encryption

Example plaintext : *M* = 127

$$C = \left(127^{(500001^2-1)/53000^2}\right) mod\ 3929183^{\ (3482^2-1)/531^2}\ mod\ 3929183$$

$$C = ((127^{(89)}) mod\ 3929183)^{(43)}\ mod\ 3929183$$

$$C = (3002502)^{43}\ mod\ 3929183$$

$$C = 1364038$$

Decryption

$$M = C^d\ mod\ z$$

$$M = 1364038^{356265}\ mod\ 3929183$$

$$M = 127$$

3. Results and Discussion

At this stage, the researcher will show the results of combining the Rabbit Stream and Enhanced Dual RSA algorithms using the Python programming language. Where the text data to be sent is first encrypted with the Rabbit Stream algorithm, which will later produce ciphertext. Furthermore, the key from the Rabbit Stream algorithm, whose key has been determined over 128 bit, is encrypted with the Enhanced Dual RSA algorithm, which produces the cipherkey. So that the ciphertext and cipherkey will be contributed to the recipient simultaneously. To return the encrypted text data, the recipient decrypts the cipherkey using the Enhanced Dual RSA private key algorithm, which generates a Rabbit Stream key. Then the ciphertext is decrypted using the Rabbit Stream key algorithm to produce plaintext. The specifications of the laptop used are as follows:

Processor	: 2.9 GHz *Intel Core* i7
Memory	: 8 GB, 1600 MHz, DDR3
SSD	: 500 GB
Operating system	: *macOS Catalina*

3.1. Program view with test data

Key : DEVANKA SARUMAHA
Plaintext : Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa.
Length of character plaintext: 100

In Figure 1, the results of the encryption and decryption process will be shown by combining the Rabbit Stream algorithm and the Enhanced Dual RSA algorithm. The processing time will be displayed automatically when the encryption and decryption process is complete.

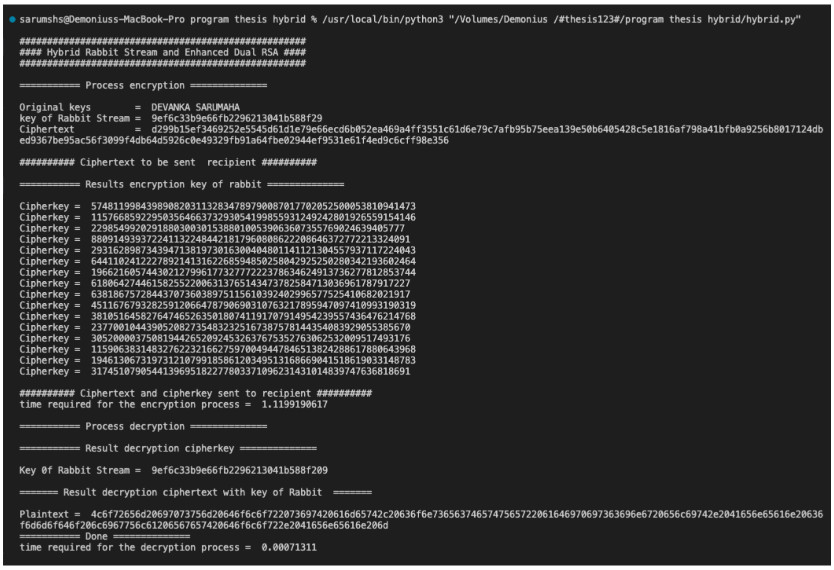

Figure 1. Encryption and Decryption Results

Key : DEVANKA SARUMAHA
Plaintext : Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec qu
Length of character plaintext: 200

The Figure 2 shows the process of encrypting text data using the Rabbit Stream algorithm and encrypting the Rabbit key with an Enhanced Dual RSA public key. The processing time will be displayed automatically when the encryption and decryption process is complete.

Figure 2. Encryption and Decryption Results

Key : DEVANKA SARUMAHA
Plaintext : Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec.
Length of character plaintext: 300

The Figure 3 shows an encryption and decryption process with a hybrid schema where the encoded text data can be restored to the original message. The runtime will be displayed automatically when the encryption and decryption process is complete.

Figure 3. Encryption and Decryption Results

Key : DEVANKA SARUMAHA
Plaintext : Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdieta
Length of character plaintext: 400

In Figure 4, the results of the encryption and decryption process will be shown by combining the Rabbit Stream algorithm and the Enhanced Dual RSA algorithm. The processing time will be displayed automatically during the encryption and decryption process.

Figure 4. Encryption and Decryption Results

From the appearance of the encryption program above that uses the same key with different plaintext character lengths, it can be seen that the resulting ciphertext, if converted into text, will show results that cannot be read or understood, and the Rabbit key encryption using the Enhanced Dual RSA algorithm produces a cipherkey that is increased many times. In the display of the decryption program, it can be seen that ciphertext that cannot be understood can be returned to the original plaintext without reducing or adding to the contents of the message characters. From the results of the encryption and decryption processes, it is concluded that the time required during the encryption process is longer when compared to the time required during the decryption process.

From the appearance of programs that use the same key with different plaintext character lengths, it shows the difference in the time needed during the encryption and decryption processes. The difference in time required is shown in Table 1.

Table 1. Encryption and Decryption Time

Key	Plaintext Character Length (Character)	Encryption Time (Second)	Description Time (Second)
DEVANKA SARUMAHA	100	1.1199190617	0.00071311
	200	1.2383890152	0.0016000271
	300	1.3180060387	0.0019190311
	400	1.5302689075	0.0033252239

Note: The key used is in accordance with the provisions of the Rabbit stream algorithm, which is 128 bits or 16 characters long.

Based on the time required for data encryption and decryption, a relationship between length of charcter and runtime speed can be seen in Figure 5.

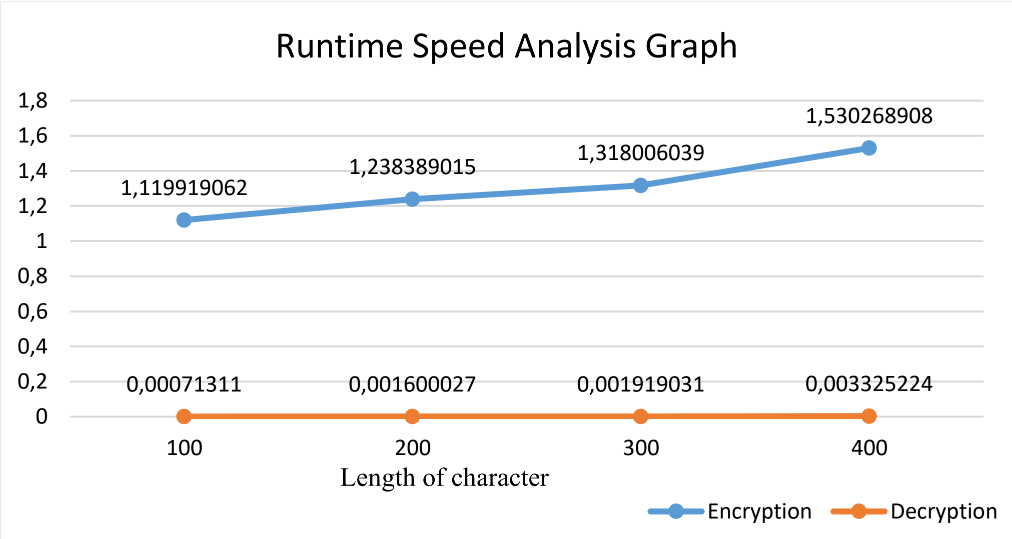

Figure 5. Processing Time Speed Graph

4. Conclusions

The study's findings suggest that using a combination of two algorithms in the encryption process generates a very large and random ciphertext that differs from the previous key. Consequently, without knowledge of the passwords for these algorithms, a hacker will need more time to compute the original message. Additionally, the Enhanced Dual RSA algorithm used in conjunction with the Rabbit Stream Algorithm generates a cipher key that is significantly more complex and challenging for cryptanalysts to decode. The encryption process requires more time compared to the decryption process, as it involves several stages. Finally, the decryption process of the ciphertext is successful in returning the original message without modifying its character content.

References

[1] D. Kusumaningsih, A. Pudoli, And I. Rahmadan, “Steganografi Metode End Of File Untuk Keamanan Data,” *J. Telemat. Mkom*, Vol. 9, No. 1, Pp. 47–55, 2017.

[2] S. Yakoubov, V. Gadepally, N. Schear, E. Shen, And A. Yerukhimovich, “A Survey Of Cryptographic Approaches To Securing Big-Data Analytics In The Cloud,” *2014 Ieee High Perform. Extrem. Comput. Conf. Hpec 2014*, 2014, Doi: 10.1109/Hpec.2014.7040943.

[3] F. Maqsood, M. Ahmed, M. Mumtaz, and M. Ali, “Cryptography: A Comparative Analysis for Modern Techniques,” Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 6, pp. 442–448, 2017, doi: 10.14569/ijacsa.2017.080659.

[4] D. P. Joseph and M. Krishna, “Cognitive Analytics and Comparison of Symmetric and Asymmetric Cryptography Algorithms,” vol. 6, no. 3, pp. 51–56, 2015.

[5] M. A. Habib, M. Ahmad, S. Jabbar, S. H. Ahmed, And J. J. P. C. Rodrigues, “Speeding Up The Internet Of Things: Leaiot: A Lightweight Encryption Algorithm Toward Low-Latency Communication For The Internet Of Things,” Ieee Consum. Electron. Mag., Vol. 7, No. 6, Pp. 31–37, 2018, Doi: 10.1109/Mce.2018.2851722.

[6] M. A. Budiman, M. Y. Saputra, And Handrizal, “A Hybrid Cryptosystem Using Vigenère Cipher And Rabin-P Algorithm In Securing Bmp Files,” J. Comput. Appl. Informatics, Vol. 4, No. 2, Pp. 2016–2017, 2018, [Online]. Available: Https://Www.Data-Science.Ruhr/About_Us/.

[7] F. Akhyar, S. M. Nasution, And T. W. Purboyo, “Rabbit Algorithm For Video On Demand,” Apwimob 2015 - Ieee Asia Pacific Conf. Wirel. Mob., Pp. 208–213, 2016, Doi: 10.1109/Apwimob.2015.7374978.

[8] B. S. Muchlis, M. A. Budiman, And D. Rachmawati, “Teknik Pemecahan Kunci Algoritma Rivest Shamir Adleman (Rsa) Dengan Metode Kraitchik,” J. Penelit. Tek. Inform. E-Issn 2541-2019, P-Issn 2541-044x, Vol. 2, No. 2, Pp. 49–64, 2017, [Online]. Available: Http://Jurnal.Polgan.Ac.Id/Index.Php/Sinkron/Article/View/75.

[9] Y. Funabashi, A. Shibata, S. Negoro, I. Taniguchi, And H. Tomiyama, A Dynamic Programming Algorithm For Energy-Aware Routing Of Delivery Drones, Vol. 13. 2020.

[1] F. Maqsood, M. Ahmed, M. Mumtaz, and M. Ali, “Cryptography: A Comparative Analysis for Modern Techniques,” Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 6, pp. 442–448, 2017, doi: 10.14569/ijacsa.2017.080659.

[2] D. P. Joseph and M. Krishna, “Cognitive Analytics and Comparison of Symmetric and Asymmetric Cryptography Algorithms,” vol. 6, no. 3, pp. 51–56, 2015.

[3] M. Boesgaard, M. Vesterager, and E. Zenner, “The rabbit stream cipher,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 4986 LNCS, pp. 69–83, 2008, doi: 10.1007/978-3-540-68351-3_7.

[11] O. G. Abood And S. K. Guirguis, “A Survey On Cryptography Algorithms,” Int. J. Sci. Res. Publ., Vol. 8, No. 7, 2018, Doi: 10.29322/Ijsrp.8.7.2018.P7978.

[12] S. Verma And D. D. Garg, “Efficient Rsa Variants For Resource Constrained Environment,” Secur. Commun. Ne*tworks*, No. August, Pp. 1–22, 2012.