PARALLEL ENCRYPTION WITH DIGIT ARITHMETIC OF COVERTEXT MODEL USING DUPLICATED COVERTEXT

Hari Murti[1*], Edy Supriyanto[2], Rara Sriartati Redjeki[3], Eka Ardhianto[4]
Sistem Informasi, Universitas Stikubank, Semarang, Indonesia[1,2,3]
Teknik Informatika, Universitas Stikubank, Semaran, Indonesia[4]
E-mail address: harimurti@edu.unisbank.ac.id[1], edysupriyanto@edu.unisbank.ac.id[2], raraartati@edu.unisbank.ac.id[3], ekaardhianto@edu.unisbank.ac.id[4]

Received: 06, September, 2022 ***Revised: 09, December, 2022*** ***Accepted: 09, December, 2022***

ABSTRACT

Steganography is the art and science of writing hidden messages or hiding messages in such a way that, apart from the sender and the recipient, no one knows or is aware that there is a secret message (it does not appear that there is a hidden message. Furthermore, public key cryptography is the type of cryptography that using two interrelated keys, namely the public and private keys. Cryptography and steganography techniques have an important role in securing information. Parallel Encryption with Digit Arithmetic of Covertext (PDAC) encryption model adopts cryptography and steganography for securing messages. The accuracy of the number of covertext in the PDAC affects whether the PDAC works. If the number of PDAC covertext does not match, then the encryption process will be imperfect. This study aims to close the PDAC gap. The proposed model adopts the repeated use of covertext. Covertext loops applied in PDAC can process information security. Another advantage obtained is that this modification makes it easier for users to use PDAC without having to meet the minimum covertext requirement.

Keywords: Covertext, PDAC, Encryption, Key.

1. INTRODUCTION

Information security is important to protect information from unauthorized parties. Information security is carried out through encryption and decryption processes in the context of the field of cryptography (Nahar & Chakraborty, 2020). Cryptography aims to maintain the authenticity of the information content by randomizing the information content so that it becomes difficult to translate (Ardhianto et al., 2021). Decryption aims to return the information to its original form that can only be read by authorized parties (Ardhianto, 2020).

Another information security technique is Steganography. Steganography and Cryptography come from Greek. Steganography comes from the words Steganos, which means "hidden", and Graphien, "to write". Cryptography comes from the word Kryptos which means "secret" and Graphein, "to write" (Ardhianto et al., 2020; Telaumbanua & Zebua, 2020) Both have the same function but have different purposes, Steganography hides messages by inserting each digit of the message into another non-secret message called a cover, so that no one knows that there is a secret message in the other message. Steganography saves messages into the cover without changing the cover file format (N et al., 2007). Cryptographic techniques hide messages by

masking or scrambling messages that have other meanings or making the original messages meaningless. The advantage of steganography over cryptography is that the results of changing messages do not arouse suspicion (Handoko, Ardhianto, Hadiono, et al., 2020). To strengthen the security of information, the integration of Steganography and Cryptography has often been proposed. The merger is considered to make it difficult for a third party known as the "man in the middle".

2. THEORY

The PDAC (Parallel Encryption with Digit Arithmetic of Cover Text) encryption model is a mathematical calculation technique and parallel concept for a text-based steganography approach (Handoko, Ardhianto, & Supriyanto, 2020; Kataria et al., 2013). PDAC uses Steganography to encrypt messages, with the stage of changing the character digits in the message into ASCII code digits, ASCII code is converted into binary code, as well as the covertext characters used. To encrypt a message, PDAC requires 1 character as covertext to generate 2 encryption keys. This is assessed as the PDAC covertext capacity is n/4, this means that every 1-character covertext can encrypt as many as 4 characters. After the covertext is converted to ASCII code to generate the encryption key, a mathematical calculation process is required SUM (addition) between the 2-digit numbers in the ASCII code and SUB (subtraction) between the 2-digit numbers in the ASCII code. The results of SUM and SUB are added 10 each to generate the encryption key (Kataria et al., 2013). The PDAC encryption process uses XOR operations between each digit of the plaintext character with the key. The result in the form of ciphertext is obtained from combining covertext with encryption results. Figure 1 shows the PDAC encryption process.

Figure 1 shows the PDAC encryption process with a plaintext length of 4 characters (KODE) so that only 1 character (R) of covertext is required. In this process, each plaintext character can be encrypted and produce ciphertext (R__TQ). This means that the number of covertext requirements is in accordance with the covertext capacity, which is n/4.

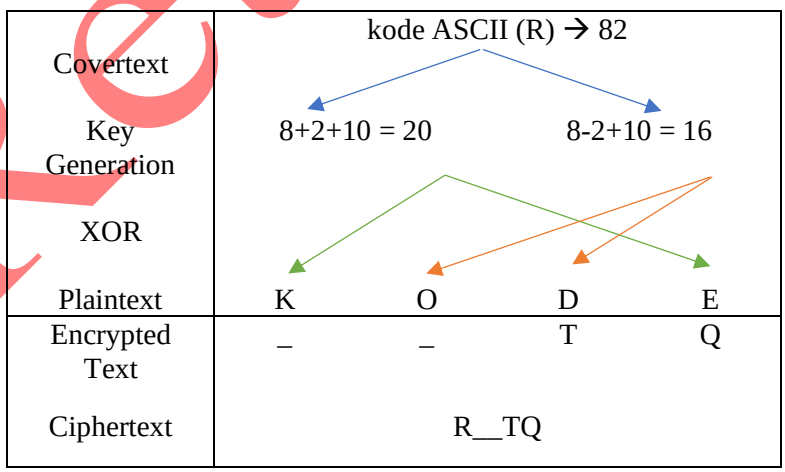

Figure. 1. PDAC Encryption Process.

PDAC has evolved. One of the evolutions of PDAC is in terms increasing its covertext capacity. Gaur (Gaur & Sharma, 2015) developed PDAC into New PDAC. This model improves the covertext capacity from n/4 to n/6. The number n/6 means that every 1 covertext can be used to process a maximum of 6 plaintext characters. Another advantage obtained is

that there is a decrease in the size of the resulting ciphertext file. The development of covertext capacity is also carried out by Handoko (Handoko, Ardhianto, & Supriyanto, 2020) which can increase the covertext capacity up to n/8, this means that every 8 plaintext characters only 1 covertext is needed.

Although the PDAC has changed to increase capacity, the number of covertext requirements must be adjusted to the total length of the plaintext. If the need for covertext does not match the length of the plaintext, then the encryption process does not run perfectly. Figure 1 shows that 4 plaintext characters require 1 covertext character, this corresponds to the n/4 covertext capacity. If there are 6 plaintext characters, then 2 covertext is needed, because 6/4 = 1.5 2 characters. If the covertext provided is only 1 character, then 2 characters that are not processed, namely D and E. Figure 2 illustrates the encryption process with the number of covertexts that do not match the length of the plaintext.

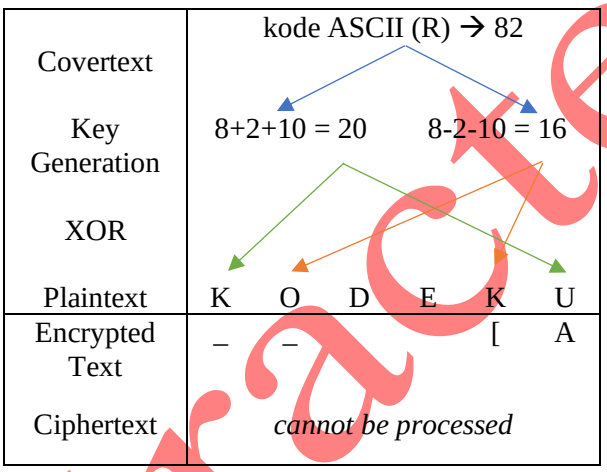

Figure. 2. The PDAC encryption process with an incorrect amount of covertext

Figure 1 and Figure 2 show the different processes that occur in PDAC. In Figure 2, it can be seen that the gap that occurs in PDAC is when the covertext requirement is less than n/4 then there are still some plaintext characters that are not encrypted so the encryption process cannot run perfectly. This article discusses how to solve the problem if in securing information using PDAC there is a covertext of less than n/4 so that the information security process can be carried out on all the plaintexts.

3. METHOD

The gap in the PDAC encryption process that was found was when the number of covertext characters did not meet the n/4 number of plaintext characters required as shown in Figure 2. To fix the gaps found, it was necessary to use the proposed covertext repetition technique, such as using the key in the Vigenere algorithm (Nofiyanto et al., 2014; Qowi & Hudallah, 2021; Telaumbanua & Zebua, 2020). Figure 3 shows the flowchart of the PDAC modification using the key repetition process.

P-ISSN : 2088-6705
E-ISSN : 2621-5608

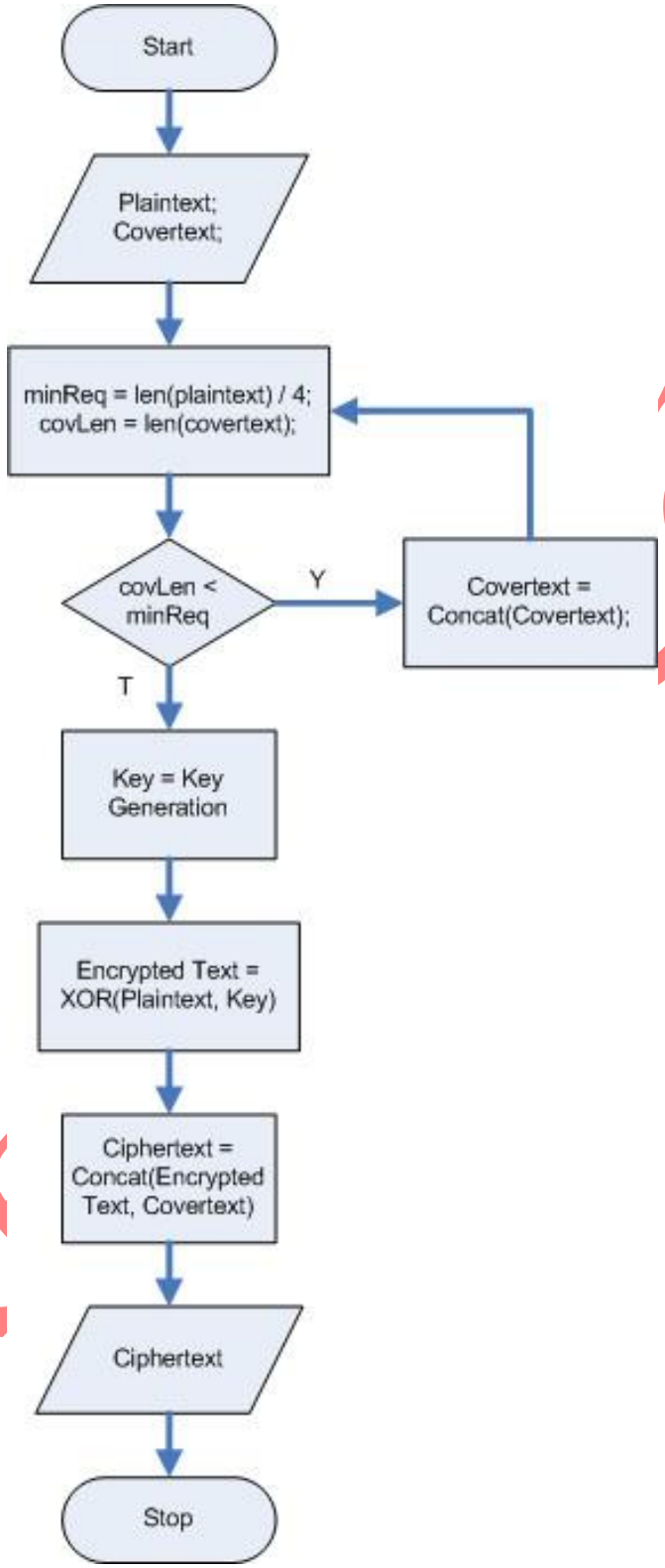

Figure 3. PDAC Encryption Process with covertext repetition.

Figure 3 shows a flowchart of the modification of the PDAC encryption model by adopting the encryption technique of the Vigenere Algorithm with key repetition according to the number of plaintext characters. This adoption is placed by looking at the difference between the minimum need for covertext and the amount of covertext entered. The PDAC encryption model

requires a covertext of at least 25% of the plaintext length. If the difference value has not reached zero or less than zero, then the covertext is repeated and connected until it meets the minimum required length. Thus, the owner of the message does not need to re-enter the covertext.

The key generation process follows the standard process according to the PDAC. This key generation requires covertext as input which is then added and subtracted between the digits of the ASCII code of each covertext. The results obtained in the form of decimal numbers are used as keys in the encryption process. The encryption process is carried out using the XOR operator between the key and the plaintext character that is adjusted to the specified arrow path. This operation corresponds to the previous version of the PDAC encryption process. The encryption results obtained are combined with covertext and produce ciphertext.

4. RESULTS AND DISCUSSION

This experiment has conducted an experiment using 2 texts as examples of plaintext and covertext as shown in table 1. The process carried out is encryption and decryption using the proposed PDAC model. The encryption process is carried out according to the flow shown in Figure 3. The decryption process in this experiment is in accordance with the decryption process in the initial version of the PDAC model. Figure 4 shows the encryption process of plaintext 1 and figure 5 shows the encryption process of plaintext 2.

Table 1. PDAC Encryption Process Sample.

	Sample (1)	**Sample (2)**
Plaintext	KODEKU	KODERAHASIA
Covertext	R	RH

The encryption in Figure 4 is done using a plaintext sample 1. First, the minimum number of covertext requirements is calculated, in the example it is rounded up, which is 1.5 2. So that the covertext R is doubled by the difference between the minimum covertext requirement and the covertext inputted by the sender so that the covertext length is fulfilled by 2 characters (RR). The key issuance process is done by converting the covertext character (RR) into ASCII code (82). The addition and subtraction operations on the ASCII code digits are performed to obtain the encryption key. The addition of the number 10 is used to avoid negative results. The encryption between the plaintext and the encryption key is processed using XOR. The first key encrypts the 1st and last plaintext characters(n), the second key encrypts the 2nd character and the 2nd character from the end (n-1), and so on. Finally, the insertion of covertext characters at intervals of every 4 characters of the encryption result, in the ciphertext. In this process, the initial covertext (R) with a length of 1 character is adjusted to the minimum requirement of 2 characters, so that the covertext becomes (RR). In this experiment, the first covertext character processes 4 characters, and the second covertext processes 2 characters.

Covertext	ASCII (R) → 82
minimum requirement	6 / 4 = 1,5 ≈ 2
Number available covertext	1

P-ISSN : 2088-6705
E-ISSN : 2621-5608

(deficiency)	2 – 1 = 1
Covertext	RR
Key Generation	8+2+10 = 20 ; 8-2+10 = 16 ; 8+2+10 = 20 ; 8-2+10 = 16
XOR	
Plaintext	K O D E K U
Encrypted Text	_ _ P Q [A
Ciphertext	R__PQR[A

Figure 4. PDAC encryption covertext repetition technique with plaintext sample 1.

Covertext	ASCII (R) → 82	ASCII (H) → 72
Minimum Requirement	11 / 4 = 2,75 ≈ 3	
Number available covertext	2	
(deficiency)	3 – 2 = 1	
Covertext	RHR	
Key Generation	8+2+10 = 20 ; 8-2+10 = 16 ; 7+2+10 = 19 ; 7-2+10 = 15 ; 8+2+10 = 20 ; 8-2+10 = 16	
XOR		
Plaintext	K O D E R A H A S I A	
Encrypted Text	_ _ W J F Q \ N @ Y U	
Ciphertext	**R**__WJ**H**FQ\N**R**@YU	

Figure 5. PDAC encryption covertext repetition technique with plaintext sample 2.

The plaintext sample 2 is visualized as shown in Figure 5. The minimum requirement for covertext is 2.75 3 characters. The length of the covertext given by the sender is 2 characters (RH), so it is converted into 3 characters (RHR). The key issuance process uses arithmetic operations of addition and subtraction of the ASCII covertext code digits. The encryption process is carried out using the XOR operator. The XOR process follows the same rules as earlier versions of PDAC. To form the ciphertext, covertext is inserted in the encryption result

with a rule of every 4 characters. In this process the 1st key to the 5th key processes 2 plaintext characters and the 6th key processes 1 plaintext character.

Ciphertext	**R**__PQ**R**[A
Covertext	RR kode ASCII (R) →82
Key Generation XOR Ciphertext Decrypted Text	RR 8+2+10 = 20	8-2+10 = 16	8+2+10 = 20	8-2+10 = 16 _ _ P Q [A K O D E K U
Plaintext	**KODEKU**

(a)

Ciphertext	**R**__WJ**H**FQ\N**R**@YU
Covertext	RHR kode ASCII (R) →82 kode ASCII (H) → 72
Key Generation XOR Ciphertext Decrypted Text	RHR 8+2+10 = 20	8-2+10 = 16	7+2+10 = 19	7-2+10 = 15	8+2+10 = 20	8-2+10 = 16 _ _ W J F Q \ N @ Y U K O D E R A H A S I A
Plaintext	**KODERAHASIA**

(b)

Figure 6. Decription of PDCA cover text repetition technique

Figures 6(a) and (b) show the decryption process. Covertext is taken from the ciphertext in every 4-character sequence, character 0, 5th, 10th, and so on. The key issuance process is carried out by adding and subtracting operations between the digits of the ASCII covertext code. Followed by the decryption process using the XOR operator. The results obtained are secured information (plaintext).

By adopting the covertext repetition process in PDAC, it provides an advantage in the process of securing information when the number of covertext inputted by the sender does not meet the minimum number of covertext requirements required for the PDAC encryption process. Thus, security using PDAC can run completely.

5. CONCLUSIONS AND SUGGESTIONS

Based on the experimental results, it can be concluded that to encrypt plaintext characters with an amount that is not in accordance with the covertext provisions, namely n/4, it can be done with covertext loops to encrypt all plaintext, another advantage obtained is information security when the number of covertext inputted by the sender does not meet the number minimal need for covertext, making it easier for users to be able to directly use the encryption process without the need to manually calculate the number of covertext characters that need to be used to encrypt and shorten the encryption time when the sender is in a hurry to secure information.

REFERENCES

Ardhianto, E. (2020). Improvement of Steganography Technique: A Survey. *1st International Multidisciplinary Conference on Education, Technology, and Engineering (IMCETE 2019)*, 289–292. www.scimagojr.com.

Ardhianto, E., Handoko, W. T., Murti, H., & Redjeki, R. S. A. (2021). Encryption with Covertext and Reordering using Permutated Table and Random Function. *2021 2nd International Conference on Innovative and Creative Information Technology, ICITech 2021*. https://doi.org/10.1109/ICITech50181.2021.9590171

Ardhianto, E., Trisetyarso, A., Suparta, W., Abbas, B. S., & Kang, C. H. (2020). Design Securing Online Payment Transactions Using Stegblock through Network Layers. *IOP Conference Series: Materials Science and Engineering*, *879*(1). https://doi.org/10.1088/1757-899X/879/1/012027

Gaur, M., & Sharma, M. (2015). A New PDAC (Parallel Encryption with Digit Arithmetic of Cover Text) Based Text Steganography Approach for Cloud Data Security. *International Journal on Recent and Innovation Trends in Computing and Communication* , *3*(3), 1344–1352. http://www.ijritcc.org

Handoko, W. T., Ardhianto, E., Hadiono, K., & Sutanto, F. A. (2020). Protecting Data by Socket Programming Steganography. *IOP Conference Series: Materials Science and Engineering*, *879*(1). https://doi.org/10.1088/1757-899X/879/1/012028

Handoko, W. T., Ardhianto, E., & Supriyanto, E. (2020). MODIFIKASI NEW PDAC (PARALLEL ENCRYPTION WITH DIGIT ARITHMETIC OF COVER TEXT). *SENDIU 2020*, 55–59.

Kataria, S., Singh, B., Kumar, T., & Shekhawat, H. S. (2013). PDAC (Parallel Encryption with Digit Arithmetic of Cover Text) Based Text Steganography. *Proc. of Int. Conf. on Advances in Computer Science, AETACS*.

N, M. K., Jayaramu, H. S., Kurian, M. Z., & Shiva kumar, K. B. (2007). FPGA Implementation of Vigenere Cipher Method Based on Colour Image Steganography. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO*, *3*(4), 9051–9057. www.ijareeie.com

Nahar, K., & Chakraborty, P. (2020). A Modified Version of Vigenere Cipher using 95×95 Table. *International Journal of Engineering & Advanced Technology (IJEAT)*, *9*(5), 1144–1148. https://doi.org/10.35940/ijeat.E9941.069520

Nofiyanto, N., Hamzah, hamzah, & Surbakti, H. (2014). SHORT MESSAGE ENCRYPTION APPLICATION DEVELOPMENT USING VIGENERE ALGORITHM UTILIZING EULER'S NUMBER ON ANDROID SMARTPHONE. *Jurnal Teknologi Informasi* , *9*(27), 81–92.

Qowi, Z., & Hudallah, N. (2021). Combining caesar cipher and hill cipher in the generating encryption key on the vigenere cipher algorithm. *Journal of Physics: Conference Series*, *1918*(4), 1–6. https://doi.org/10.1088/1742-6596/1918/4/042009

Telaumbanua, F., & Zebua, T. (2020). Modifikasi Vigenere Cipher Dengan Pembangkit Kunci Blum Blum Shub. *KOMIK (Konferensi Nasional Teknologi Informasi Dan Komputer)*, *4*(1). https://doi.org/10.30865/komik.v4i1.2646