**Advance Sustainable Science, Engineering and Technology (ASSET)**
**Vol. 6, No.1, January 2024, pp. 0240107-01 ~ 0240107-08**
**ISSN: 2715-4211 DOI:** https://doi.org/10.26877/asset.v6i1.17332

# A Good Result for Blowfish Image Encryption Based on Stepic

**Anis Putma Cahyani[1*], Ajib Susanto[1]**

[1]University of Dian Nuswantoro, Jl. Imam Bonjol 207, Semarang, Central Java 50131, Indonesia

*111202214375@mhs.dinus.ac.id

**Abstract**. Information security is important in the era of evolving communications. Security methods are needed to protect data. Strong cryptographic algorithms, such as Blowfish, developed by Bruce Schneier in 1993, offer fast and reliable encryption. Blowfish uses 64 bit blocks and key lengths between 32 and 443 bits. This algorithm is famous for its robustness and encryption speed, with a Feistel Network structure and 16 rounds. This research implements Blowfish with Python and integrates it with steganography to insert secret data in digital images. Evaluation involves metrics such as Mean Squared Error (MSE), Peak-Signal-to-Noise Ratio (PSNR), Average Visibility (AVA), Uniform Average Change Intensity (UACI), and Percentage Pixel Change Value (NPCR) to measure quality and reliability encryption and decryption process by Blowfish.

**Keywords**: Blowfish, MSE, AVA, PSNR, NPCR, UACI

***(Received 2023-11-04, Accepted 2023-12-07, Available Online by 2023-12-10)***

## 1. Introduction

Information security plays a very important role in the current era of communication media development. In an era where data moves rapidly over digital networks, the challenge of protecting data confidentiality and integrity is increasingly complex. Security threats, such as hacking and theft of confidential information, are becoming increasingly serious. In fact, several high-profile security incidents have revealed inherent vulnerabilities in existing systems and protocols. Therefore, alternative security methods are needed to maintain data confidentiality and integrity [1]. Strong cryptographic algorithms are needed so that only authorized parties can access and understand the information [2]. One cryptographic algorithm that has been widely used and proven to be effective is Blowfish [3].

Blowfish was developed in 1993 by Bruce Schneier as an open-source encryption alternative that does not require a license. Its advantages in terms of compatibility and efficiency have been widely recognized, making it a fast and reliable encryption algorithm [4]. Blowfish is a cryptographic block cipher with a fixed block length of 64 bits and a key length that can vary between 32 to 443 bits [5]. Apart from being renowned for its resistance to attacks, Blowfish also stands out for its optimal speed of data encryption and decryption, both on hardware and various software platforms [6].

In this research, we aim to implement the Blowfish algorithm using the Python programming language. In addition, we will integrate the Blowfish algorithm with steganography, a technique that

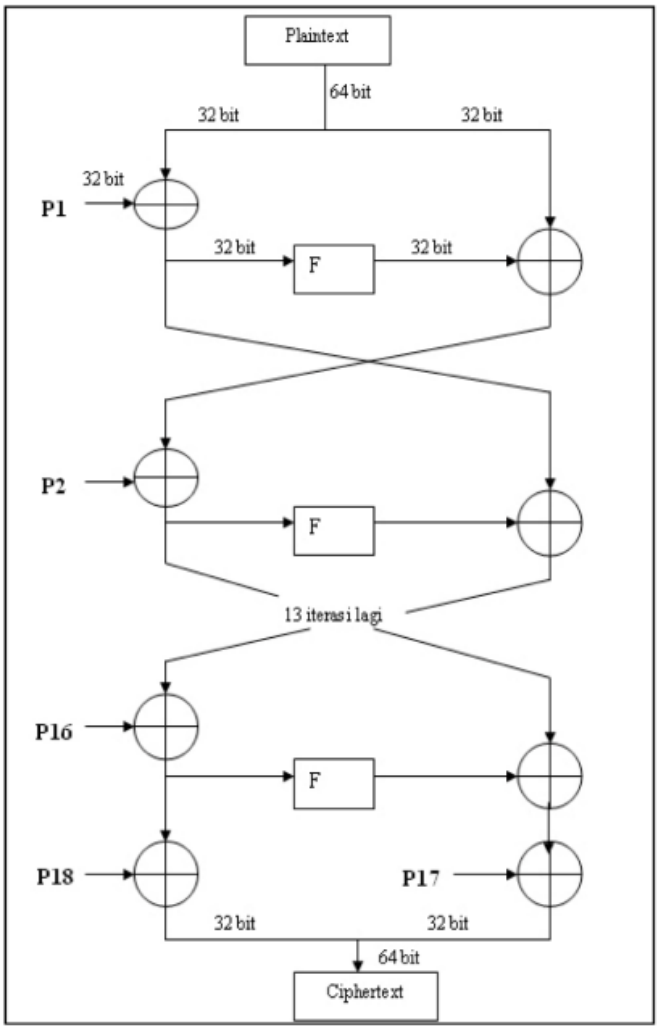
allows us to embed secret data into digital images. We will also evaluate this research by measuring the quality and reliability of the encryption and decryption process using the Blowfish algorithm. Some of the evaluation metrics that will be used include Mean Squared Error (MSE), Peak-Signal-to-Noise Ratio (PSNR), Average Visibility (AVA), Uniform Average Change Intensity (UACI), and Percentage Pixel Change Value (NPCR) [7].

## 2. Methods

### *2.1. Blowfish Algorithm*

Blowfish is a block cipher algorithm that uses blocks of the same 64-bit size during the encryption and decryption process. In this process, the message to be encrypted is divided into blocks of k bits of a fixed size, namely 64-bit [5]. The encryption process in the blowfish algorithm is carried out on data blocks with a fixed length of 8 bytes, although the key length can vary. Addition of bits (padding) will be done if there is a message size that is not a multiple of 8 bytes, so that the size of each block is the same so that with a uniform size Blowfish can encrypt data effectively [7].

Blowfish is a block cipher algorithm that has two main processes, namely key expansion and data encryption. In the key expansion process, the initial key with a maximum length of 448 bits is converted into several subkeys which are stored in an array with a total size of 4168 bytes. Meanwhile, the data encryption process consists of 16 iterations of a simple function that operates the data repeatedly with the resulting sub-keys.




0240107-02

**Figure 3.** Blowfish Scheme

In the key expansion stage, the initial key with a maximum length of 448 bits will be converted into several subkeys. These subkeys will be stored in an array with a total size of approximately 4168 bytes. The data encryption process itself consists of 16 iterations of a simple function that operates on the data with the resulting subkeys.

The Blowfish encryption process begins by dividing the data to be encrypted (X) into two parts, namely XL and XR, each consisting of 32 bits [8]. Next, a series of steps are carried out as follows:

1. Iterate 16 times, starting from i = 1 to 16.
2. At each iteration, XL is XORed with the Pi subkey. Then XR is XORed with the result of the function F(XL) and XORed with the previous XR.
3. After that, the XL and XR values are swapped.
4. After the 16th iteration, a final exchange is carried out between XL and XR.
5. Next, XR is XORed with subkey P17, and XL is XORed with subkey P18.
6. Finally, XL and XR are combined again to get the ciphertext.

This process uses addition and XOR operations on 32-bit variables. With these steps, data can be encrypted using the Blowfish algorithm

*2.2. Data Embedding with Steganography*

In this research, steganography is used for the purpose of hiding the existence of important information by inserting messages into objects or media that look harmless. This aims to make this information difficult for unauthorized third parties to detect, so that the data can be better protected. By using steganography, secret messages can be effectively hidden in objects or media that do not appear suspicious [9].

In the steganography process, two files are needed, namely the container file and the important data [10]. Container files are media where important data will be inserted. The types of media that can be used as container files vary, such as text, images, audio, or video. In this study, specifically, we use images as container files. By using an image as a container file, important data will be inserted into the pixels of the image without being visible, so that the desired information can be hidden properly.

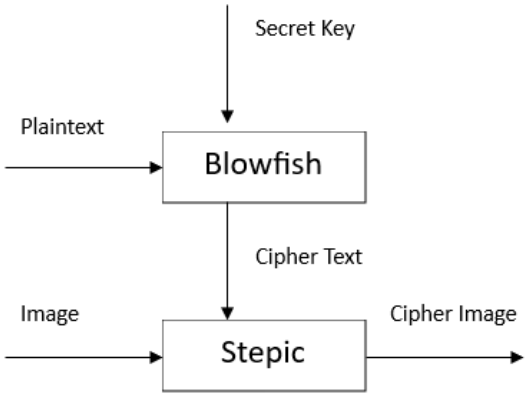



**Figure 2.** Encryption Scheme

This process involves implementing the Blowfish algorithm described previously. Once the ciphertext is encrypted, it is saved and then inserted into an image selected by the user. This hiding process aims to protect data by inserting it invisible in the image.

## 3. Results and Discussion

In this stage, testing is carried out on the system, which looks at the quality of the storage media, whether it experiences changes during the data insertion process. We here use 3 media containers in the form of images.

**Table 1.** List of Dataset

| Container Media | File Name |
|---|---|
|  | Image1 |
|   | Image2 |
|  | Image3 |

The image quality resulting from this process can be measured using MSE, AVA and PSNR. MSE is a measure of the average square error between the original image and the image containing a hidden message. PSNR is the comparison of pixel values between the image (original and stego image) produced [9]. A cryptographic algorithm will fulfill the avalanche effect if every change in one input bit causes half of all output bits to change [11].

The formula for calculating MSE is as follows: Information :

$$MSE_{AVG} = \frac{MSE_R + MSE_B + MSE_G}{X.Y} \quad [12]$$

Information :
MSE : Mean Square Error Value of the Image
XY : Dimensions of the image

The formula for calculating PSNR is as follows:

$$PSNR = 20 * \log 10 \left(\frac{255}{\sqrt{MSE}}\right) \quad [10]$$

Information :
PSNR:Image PSNR value
MSE : MSE value

After calculating the MSE and PSNR, it was found that the MSE did not change much, the highest PSNR value was 86.54 and the highest AVA value was 35.85. And also obtained an average MSE value of 0.00, Ava 34.87 and meanwhile the average PSNR value obtained was 79.22. From these results, a low MSE value means the better the quality of data insertion. On the other hand, a high PSNR value indicates that the resulting image quality is getting better. We also evaluated the image using the 3 existing images. Testing uses UACI and NCPCR where we try to encrypt images with different message length variants.

**Table 2.** MSE dan PSNR Test Result

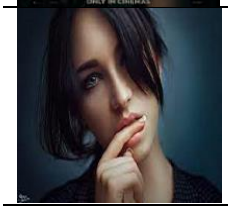
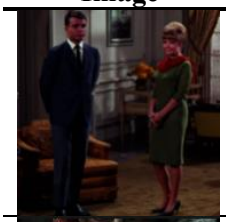
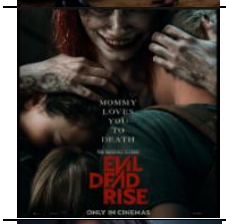
| Image | Messsage Size (byte) | MSE | PSNR | AVA |
|---|---|---|---|---|
|  | 13.716 | 0,00 | 80,94 | 35,85 |
| | 89.154 | 0,00 | 74,36 | 34,10 |
| | 185.166 | 0,00 | 71,36 | 34,66 |
|  | 13.716 | 0,00 | 86,04 | 35,85 |
| | 89.154 | 0,00 | 79,68 | 34,10 |
| | 185.166 | 0,00 | 76,59 | 34,66 |
|  | 13.716 | 0,00 | 86,54 | 35,85 |
| | 89.154 | 0,00 | 80,26 | 34,10 |
| | 185.166 | 0,00 | 77,27 | 36,66 |

**Table 3.** UACI and NPCR Test Result

| Image | Messsage Size (byte) | UACI | NPCR |
|---|---|---|---|
|  | 13.716 | 36,75 | 99,9 |
| | 89.154 | 36,76 | 99,56 |
| | 185.166 | 36,78 | 99,12 |
|  | 13.716 | 33,77 | 99,97 |
| | 89.154 | 33,77 | 99,88 |
| | 185.166 | 33,77 | 99,76 |
|  | 13.716 | 20,55 | 99,98 |
| | 89.154 | 20,55 | 99.89 |
| | 185.166 | 20,56 | 99,79 |

Judging from these results, the UACI results can be seen that when the message inserted is longer, the UACI value increases, whereas, conversely, the longer the message inserted, the lower the NCPCR value, which shows that the image changes when the message is inserted. Overall, the average UACI was 30.36 and NCPCR was 99.76. To see in more detail the changes in the image before the message is inserted with the image that has been inserted in the message. In Figure 3, where Image1 has not had a message inserted, it can be seen in the histogram that there are many spikes compared to Figure 4, where Image1 has had a message inserted, which has a different spike than Figure 3. This shows the change in the image when a message is inserted.

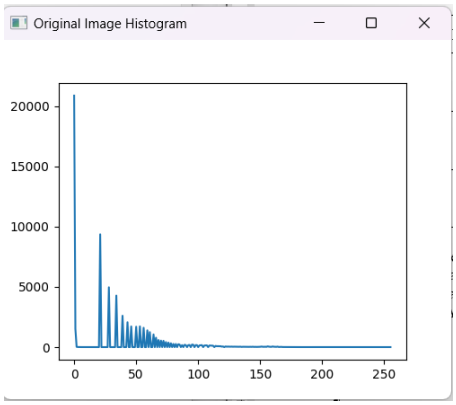



**Figure 3.** Plain Image1

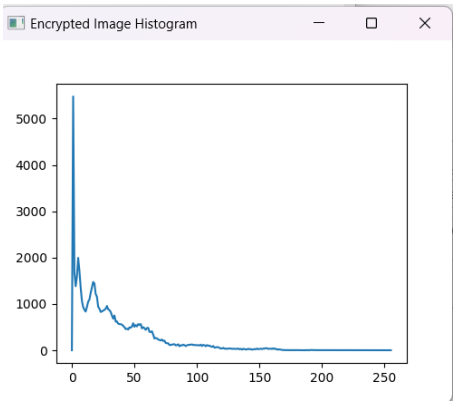



**Figure 4.** Cipher Image1

In Figure 5, the plain Image2 histogram which can be seen from the Image2 plain histogram has spikes. It can be compared with Figure 6, the Image2 image after getting the hidden message, the spikes shown in the histogram have changed.

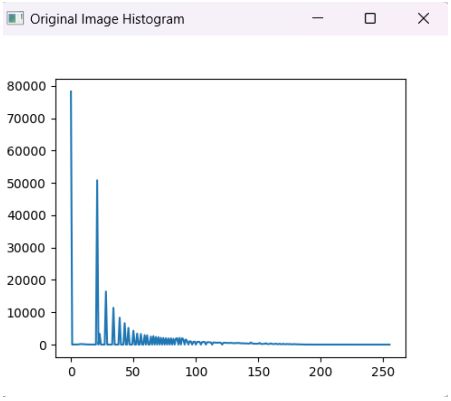



**Figure 5.** Plain Image2

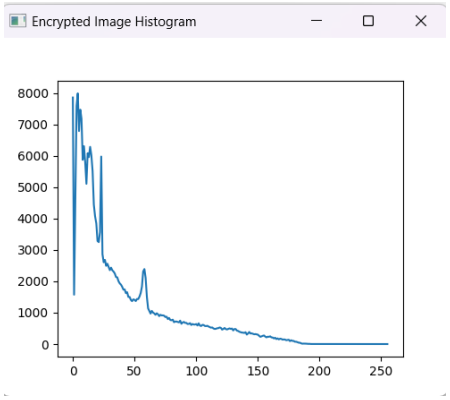



**Figure 6.** Cipher Image2

In Figure 7, it can be seen that the histogram of the original Image3 image shows a spike in the histogram value. However, if we compare it with Figure 8, which is an Image3 cipher image that has received a hidden message, we can see that the previously visible spikes in the histogram have undergone changes.

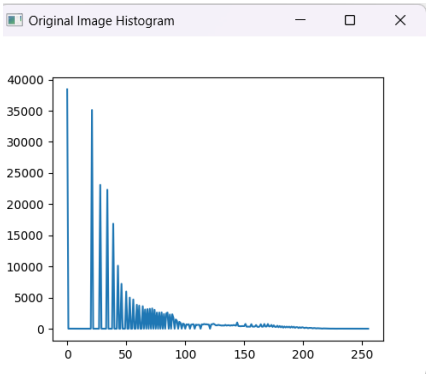



**Figure 7.** Plain Image3

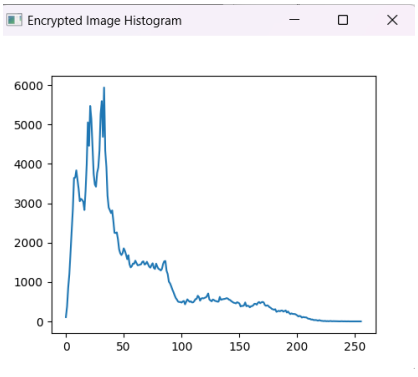



**Figure 8.** Cipher Image3

Based on tests carried out on the system using 3 image storage media, it was found that the image quality resulting from data insertion could be measured using the MSE, PSNR, UACI and NCPCR methods. The test results show that the lower the MSE value, the better the data embedding quality, while the higher the PSNR value, the better the resulting image quality. In addition, it was also found that the UACI value increased along with increasing the length of the inserted message, while the NCPCR value tended to decrease. This shows that the image changes when the message is inserted. Evaluation using the histogram also indicates a change in the histogram spike after the message insertion process.

## 4. Conclusion

From the results of this research, it can be concluded that the cryptographic method using the Blowfish algorithm which is used to insert data into images produces good image quality. This evaluation test using MSE, PSNR, AVA, UACI, NCPCR shows that this method is able to maintain image quality with a low MSE value, namely with an average of 0.00, and an average PSNR value of 79.22. This indicates that the changes that occur in the image as a result of data insertion are relatively small and not visually significant. In addition, the UACI and NCPCR test results show that the data insertion system has the desired avalanche effect where changing one input bit causes half of all output bits to change. This shows that the cryptographic system used is safe and can maintain the confidentiality of the inserted message.

## References


[1] S.- Muryanah, "MENYISIPKAN PESAN RAHASIA KEDALAM GAMBAR DENGAN METODE BLOWFISH DAN LEAST SIGNIFICANT BIT (LSB)," *JIKA (Jurnal Informatika)*, vol. 4, no. 3, Art. no. 3, Nov 2020, doi: 10.31000/jika.v4i3.2869.

[2] S. Muryanah dan S. Syam, "Aplikasi Enkripsi Kriptografi Dengan Algoritma Blowfish Dan Kompresi Huffman Dalam Security Dokumen," *Syntax : Jurnal Informatika*, vol. 10, no. 02, Art. no. 02, Nov 2021, doi: 10.35706/syji.v10i02.5541.

[3] - Muhammad Reyhan Zelvian, "KRIPTOGRAFI GAMBAR DENGAN MENGGUNAKAN ALGORITMA BLOWFISH," other, Universitas Pendidikan Indonesia, 2023. Diakses: 4 Juli 2023. [Daring]. Tersedia pada: http://repository.upi.edu

[4] Hairullah, C. Pramartha, dan I. Putra, "Aplikasi Keamanan E-Commerece Berbasis Web Menggunakan Metode Algoritma Blowfish," vol. 1, hlm. 79–87, Nov 2022.

[5] I. A. W. Arnawa, "PERBANDINGAN WAKTU ENKRIPSI ANTARA METODE ELECTRONIC CODEBOOK (ECB) DAN CHIPHER BLOCK CHAINING (CBC) DALAM ALGORITMA BLOWFISH," no. 1, 2020.

[6] B. W. Rauf, "Kombinasi Steganografi Bit Matching dan Kriptografi Playfair Cipher, Hill Cipher dan Blowfish," *JurTI*, vol. 4, no. 2, hlm. 228–233, Des 2020, doi: 10.36294/jurti.v4i2.1346.

[7] Y. P. Astuti, E. H. Rachmawanto, dan C. A. Sari, "OPTIMASI ENKRIPSI PASSWORD MENGGUNAKAN ALGORITMA BLOWFISH," vol. 15, no. 1.

[8] B. Prasetiyo, M. A. Muslim, dan H. Susanto, "Penerapan Kriptografi Algoritma Blowfish pada Pengamanan Pesan Data Teks," *tc*, vol. 16, no. 4, hlm. 358–366, Jun 2017, doi: 10.33633/tc.v16i4.1452.

[9] G. C. M. Purba dan A. ID Hadiana, "Pengamanan Citra Medis Berbasis Steganografi d an Kriptografi Dengan Menggunakan Metode End Of File Dan Advanced Encryption Standard," *INDEX*, vol. 4, no. 1, hlm. 1–9, Jul 2022, doi: 10.36423/index.v4i1.878.

[10] R. Maharani, S. H. Sitorus, dan D. Prawira, "PENGAMANAN DATA RIWAYAT PENYAKIT PADA PASIEN MENGGUNAKAN STEGANOGRAFI MOST SIGNIFICANT BIT (MSB) (Studi Kasus: Penyakit Hiv/Aids Rumah Sakit Soedarso Pontianak)," *Coding Jurnal Komputer dan Aplikasi*, vol. 8, no. 1, Art. no. 1, Jan 2020, doi: 10.26418/coding.v8i1.39207.

[11] C. Irawan, E. H. Rachmawanto, C. A. Sari, dan C. A. Sugianto, "SUPER ENKRIPSI FILE DOKUMEN MENGGUNAKAN BEAUFORT CIPHER DAN TRANSPOSISI KOLOM," 2020.

[12] G. Wibisono, T. Waluyo, dan E. I. H. Ujianto, “KAJIAN METODE METODE STEGANOGRAFI PADA DOMAIN SPASIAL,” *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 5, no. 2, Art. no. 2, Feb 2020, doi: 10.33480/jitk.v5i2.1212.