Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.754753

SEQUENTIAL AND PARALLEL EFFICIENT ENCRYPTION OF MULTIMEDIA DATA

Dr. Baby H T
Associate Professor, Government Engineering College, Mosalehosahalli, Hassan, Karnataka-573212, babygowda@gmail.com

Dr. Raghu M.E
Associate Professor, Government Engineering College, Mosalehosahalli, Hassan, Karnataka-573212, raghu.me01@gmail.com

Dr. Pallavi H V
Associate Professor, Government Engineering College, Hassan, Karnataka-573201
Karnataka-573212, bpallavi178@gmail.com

Abstract
As technology advances, encryption methods must adapt to protect sensitive information. Organizations must stay informed about emerging threats and regularly update their protocols to stay ahead. Investing in advanced encryption technologies can mitigate potential breaches. Encryption protects intellectual property, multimedia data, and personal data from cyber threats, a crucial consideration for businesses and individuals in the digital age.This work introduces four AES-based threads for encrypting and decoding multimedia data. Both sequential and parallel methods have been used for data security. Furthermore, time analyses for sequential and parallel processing algorithms have been conducted using images of various sizes and formats. The examination of the results indicates that the encryption process will be completed faster and will provide network-wide data safety in parallel method.

Keywords: Multimedia data, Sequential, Parallel, Encryption, Decryption

1. Introduction

In order to protect multimedia data from potential intrusions and maintain privacy and secrecy, encryption methods are essential. They are crucial to both individual and organizational cybersecurity strategies because they guard against intellectual property theft and guarantee safe communication routes. Encryption is a vital technique for securing digital content since it not only prevents unwanted access but also guarantees the integrity of important data. Encryption may give organizations and people peace of mind by preventing unwanted usage of multimedia files and data leakage. It is crucial in the current digital era because it makes it harder for hackers to intercept and decode critical data. By being proactive in cybersecurity, organizations can stop expensive data breaches and keep priceless assets out of the wrong hands.

2. Existing methods

Multimedia applications span across various fields, including advertising, art, education, entertainment, engineering, medicine, mathematics, business, scientific research, and spatial temporal applications. However, security concerns have arisen due to the advancement of technology. To address this, various encryption and decryption methods are required.

Encryption transforms information into a cipher text, which is unreadable to anyone except a key with special knowledge. Decryption reverses this process to plain text. Image encryption schemes now include substitution and diffusion processes. In the substitution stage, pixels are permuted without changing their value, while in the diffusion stage, pixel values are modified sequentially to spread changes to as many in the cipher image as possible[1,2,3,4,5].

Video is a large amount of visual data that requires compression algorithms for storage, transmission, and processing. Encryption techniques, such as the Data Encryption Standard (DES), Advanced Encryption Standard (AES), RSA, Triple DES (3DES), and International Data Encryption Algorithm (IDEA) and Scalable Encryption Algorithm (SEA), are used to enhance the security of multimedia data. These techniques work on bit streams of data input without distinguishing between audio, video, text, or graphics. When multimedia data is not real-time, it can be treated as a regular binary stream and conventional techniques can be applied. However, achieving security for multimedia data can be challenging when various constraints exist[6,7,8,9,10].

The video scrambling method performs distortion of the signal in the frequency domain or permutation of the signal in the time domain by using filter banks or frequency converters. However, this system offers

less protection, and sophisticated computers may readily crack this strategy. As long as the image file is compatible with the system, the program is fairly straightforward and quick to use.

Compression and encryption are used in the selective video encryption approach. This approach can handle real-time audio and video data effectively. With this procedure, just the most crucial coefficients from the last or intermediate stages of the compression process are chosen, and those coefficients are then encrypted. Less significant coefficients are not encrypted. It alludes to performance competence.

3. Proposed methodology

Multimedia data encryption and decryption employ the AES algorithm, a symmetric cryptography technique that iteratively repeats the same procedures. It is a reversible secret key encryption technique that uses a fixed number of bytes and does the same operations in reverse order. In this study, an approach for AES encryption and decryption is proposed. The inputs for the encryption process include image, video, and audio. There are four steps in the process: adding round keys, mixing columns, moving rows, and replacing bytes. The same encryption procedure is used for decryption, but the keys are used in reverse order. Figure 1 shows the general block diagram of symmetric encryption and Decryption process.

Figure 1: General Block Diagram of Multimedia Data Encryption and Decryption

4. Implementation

4.1 Encryption and Decryption of an Image

A user-provided plain picture is encrypted using the AES technique. It entails interpreting the picture, executing the encryption algorithm, and getting the password. Afterwards, the encrypted image is added to the original image and separated into blocks based on block size. Encrypting the original picture, deleting extra bytes, and using the CBC mode method constitute the decryption process, which is the opposite of the encryption process. The picture that has been decrypted is stored back at the path.Figure 2 shows the flow chart of Encryption and Decryption of an image.

4.2 Encryption and Decryption of an Image using Threads

The process of encrypting and decrypting an image using threads involves splitting the input image into equal parts and assigning each part to a thread. The encryption is performed simultaneously using the AES algorithm, and the encrypted images are saved to a specific location. Threads are used to increase performance, and the decrypted images are then merged into the original image. Figure 3 shows the flow chart diagram of the Encryption and Decryption of an Image using Threads.

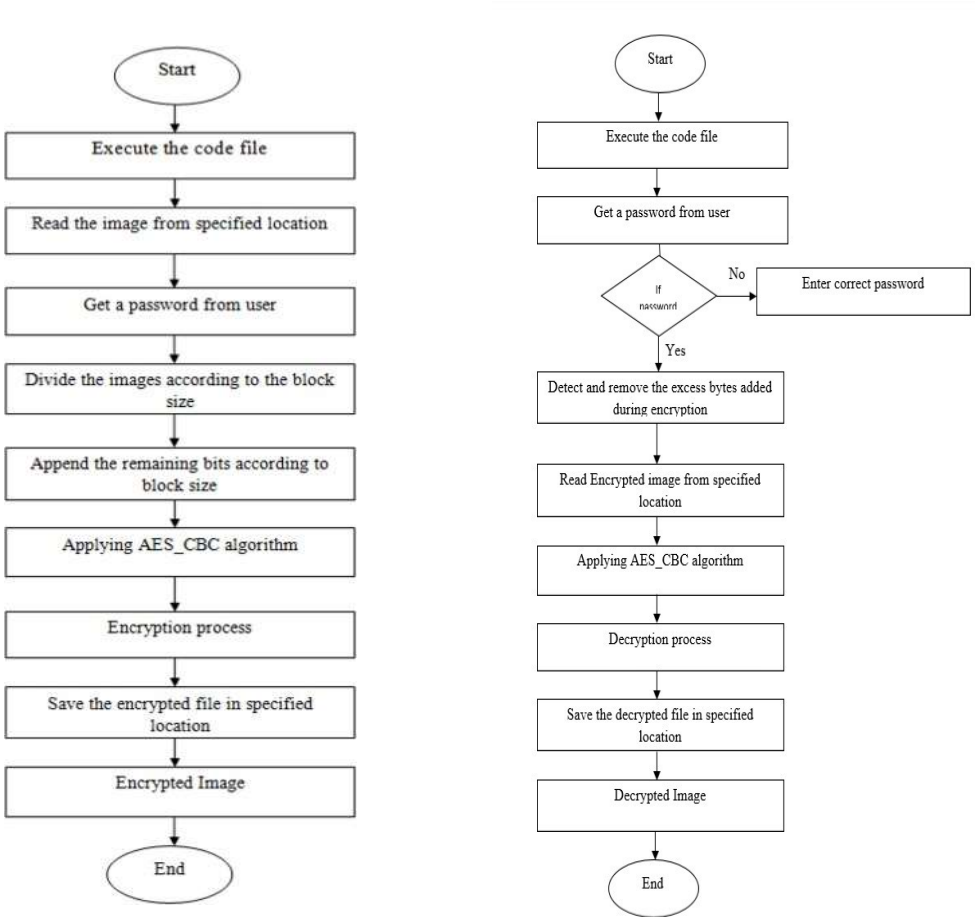

Figure2: Encryption and Decryption process of an image using AES

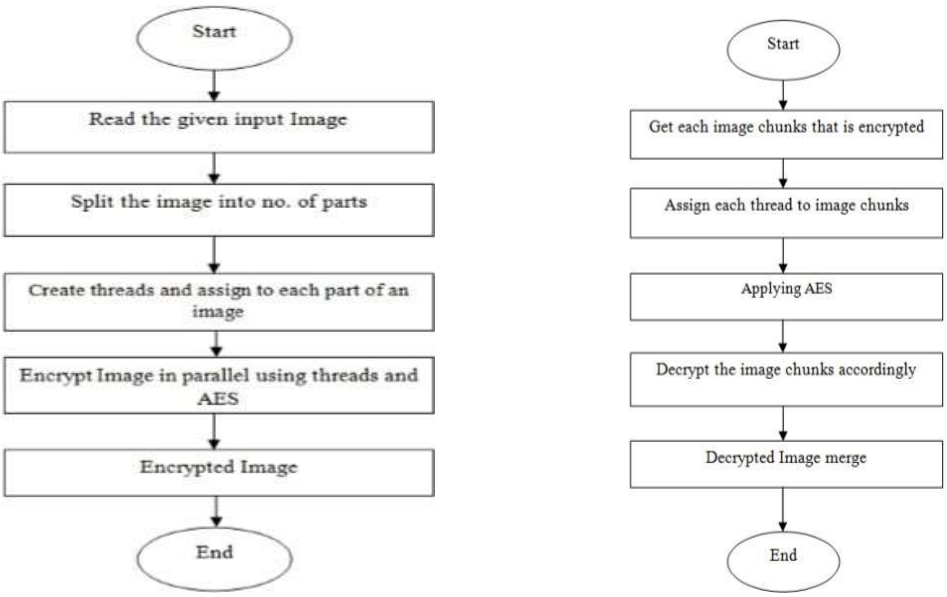

Figure 3: Encryption and Decryption of an image using threads

4.3Encryption and Decryption of Video

Step1

Video encryption involves two steps: frame extraction and encryption and decryption. The CV2 package imports a video file, extracts frames using CV2.video capture, and writes them to a path. If the video has an audio file, the audio procedure is implemented in step 2.

Step 2

The encryption and decryption process is described in Step 2. The .glob package is used to extract file names, and frames are opened as images. The image is encrypted by AES in CBC and decrypted using AES in CBC decrypt. Every file is transformed into a video using ffmpeg, and the decrypted image is saved in a location. Avconv is used to combine audio and video.

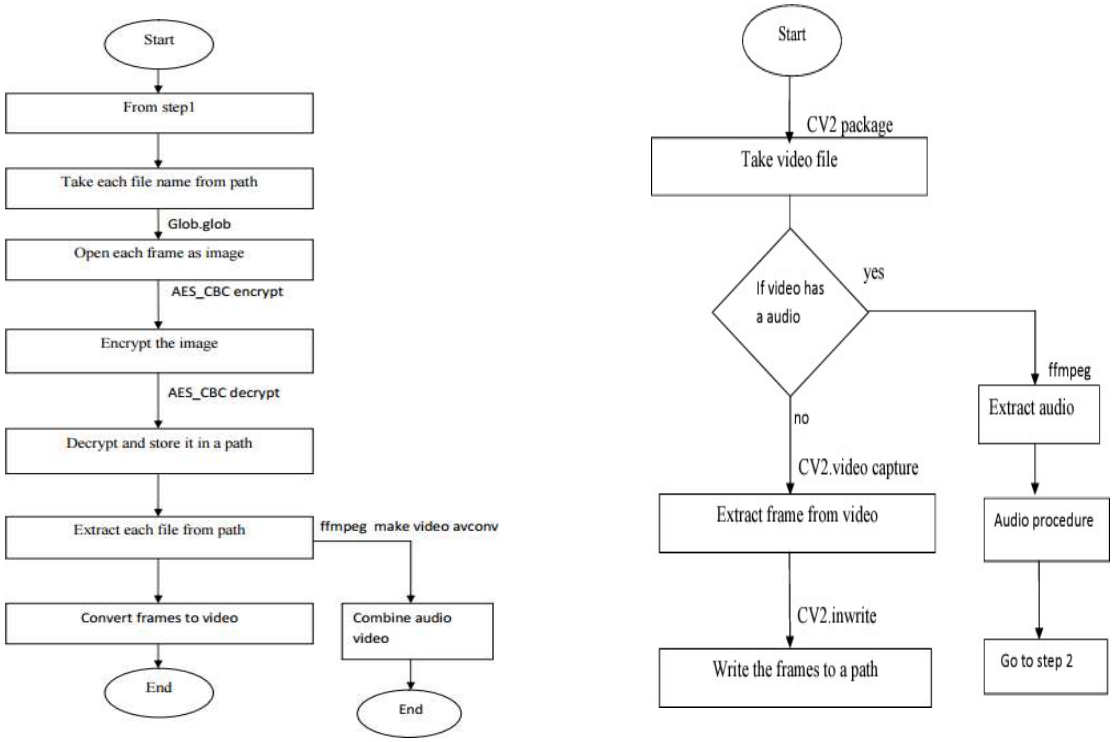

Figure 4: Encryption and Decryption of the Video using AES

4.4Encryption and Decryption of the Video using Threads

The video encryption and decryption process is performed in parallel using threads. If the video has an audio file, the audio file is encrypted and decrypted, and the video and audio are combined using avconv. If the video doesn't have an audio file, frames are divided and assigned to each thread. Frames are then encrypted and decrypted using ffmpeg. Video is extracted from

each thread and joined using ffmpeg. The original video file is then combined. The procedure is illustrated in Figure 5.

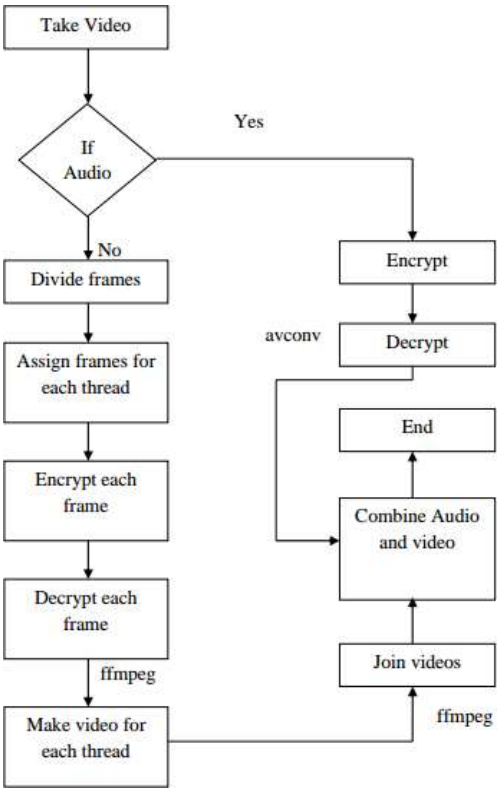

Figure 5: Encryption and Decryption of an Audio and video using Threads

5 Results:

5.1SequentialEncryptionanddecryptionofan image file

- The window seen in figure 6 is used to both initiate and end the image cryptography process. It assists in choosing an image file for encryption and decoding.

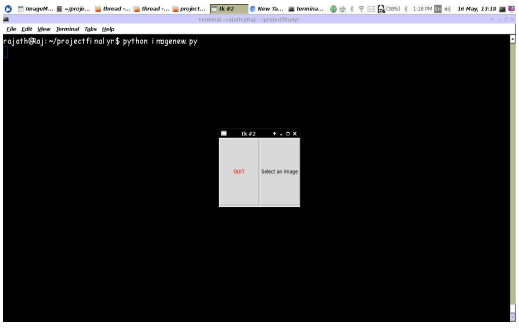

Figure 6: Menu Screen

- After choosing the "select an image" option, users must choose an input image file from the designated place, as seen in figure 7

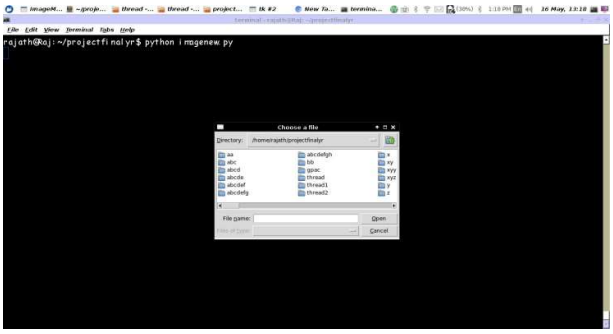

Figure 7: Selection of an Image file from the specific location

• As seen in figure 8 a password must be entered in order to encrypt and decrypt the input image file once it has been selected from the designated location. In this approach, the input image file's encryption and decryption processes need the same password to be supplied; if not, the operation will terminate by delivering an incorrect password.

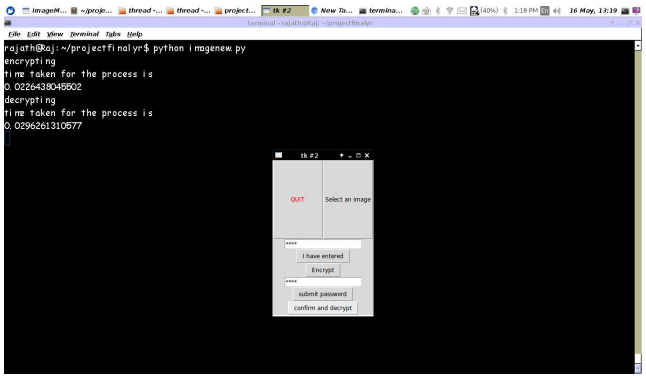

Figure 8: Entering password for encryption and decryption of an image

• The encryption of an input picture file is accomplished by using the Encrypt button, as seen in Figure 9. This initiates the encryption process, shows the resultant encrypted image in Figure 10, and indicates how long it will take to complete.

Figure 9: Input image

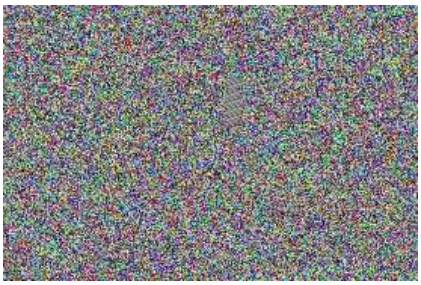

Figure 10: Encrypted image

• A designated place is used to store the encrypted image. To decrypt an encrypted image, use the "Decrypt" button. This will launch a command that will specify the method to be followed

and how long it will take. then presents an image file that has been encrypted, as seen in figure 11

Figure 11: Decrypted image

5.2 Parallel Encryption and Decryption of an Image file using Threads

A plain image file figure 12 is used in the thread-based encryption procedure, which divides the image into a number of equal portions figure 13. Every component has its own thread, and at the same time, each part of an image is encrypted in parallel using distinct threads figure 14

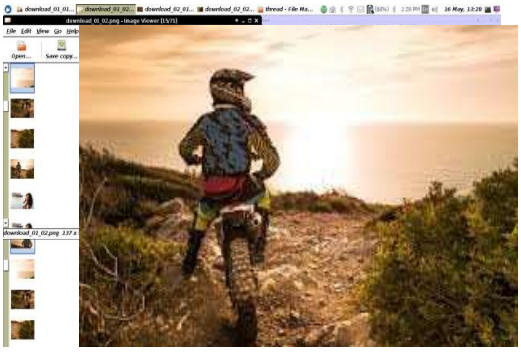

Figure 12: Input image

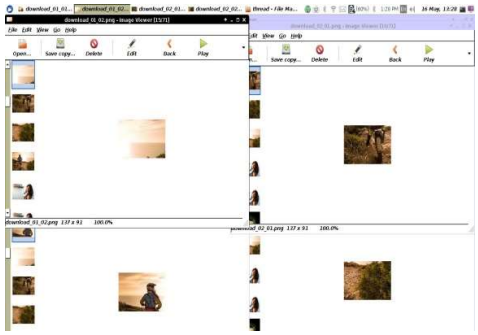

Figure 13: Input image divided into slices

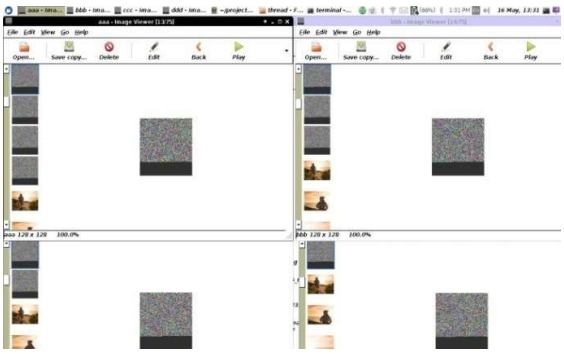

Figure 14: Encrypted image slices

The procedure for employing threads to decrypt an encrypted image is depicted in figure 15. To produce a decrypted image file, the splattered encrypted image must be decoded in parallel using each thread independently at the same time. Eventually, all of the threads must be connected together to obtain the original image as in figure 16.

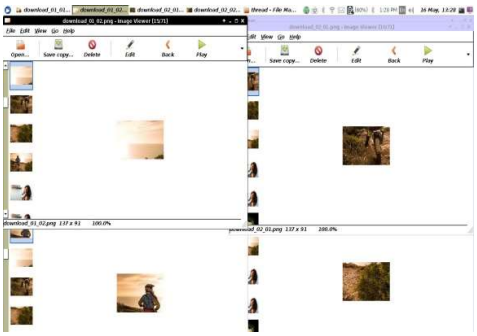

Figure 15: Decrypted image slices

Figure 16: Output image

5.3 Sequential Encryption and Decryption of a Video

The input video file that has to be taken for the encryption and decryption procedure is shown in figure 17 Following selection, the input video file will be separated into the number of frames, as figure 18 illustrates.

Figure 17: Input video

Following the splitting of the video file into its component frames, each frame must be encrypted independently and saved in the designated place, as indicated in figure 19. Next, as indicated in figure 19, every encrypted frame must be decoded independently and saved in a designated place.

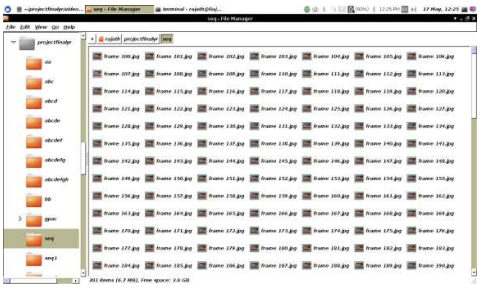

Figure 17: Video file is divided into frames frames

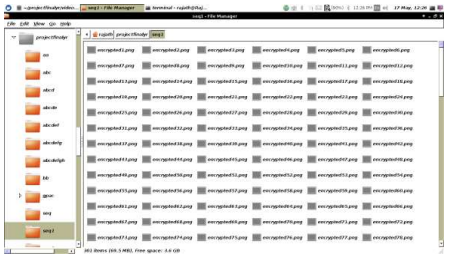

Figure 18: Encrypted video

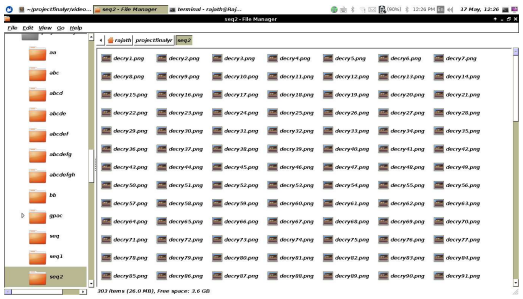

Figure 19: Decrypted video frames

Next, as seen in figure 20, the decrypted frames must be added together to obtain the original input video file.

Figure 20: Output video

5.4 Time Analysis

The start and end times of the process determine how long encryption and decryption take. When the size of the picture rises, sequential approaches become slower; in contrast, thread-based encryption and decryption yields better performance. Table 1 illustrates how half of the execution time can be saved by using threads instead of sequential techniques.

Table 1: Time analysis for sequential encryption and decryption with conversion format (in sec)

Image size	Dimensions	Conversion from Jpeg to png (time)	Conversion from Jpeg to bmp (time)	Conversion from Jpeg to jpeg(time)
7.6 MB	5184*3456	22.8224	4.1691	2.7019
9.9MB	5196*3464	20.8639	4.2870	3.9420
58.5KB	960*639	0.5414	0.1212	0.1187

Table 2: Time analysis for parallel encryption and decryption using threads with conversion format (in sec)

Image size	Dimensions	Conversion from Jpeg to png (time)	Conversion from Jpeg to bmp (time)	Conversion from Jpeg to jpeg(time)
7.6 MB	5184*3456	0.041740	0.0403	0.0345
9.9MB	5196*3464	0.3676	0.0969	0.089
58.5KB	960*639	0.037	0.0452	0.037

Tables 1 and 2 illustrate how long it roughly takes to encrypt and decode images with different formats and the same size. However, there are variations in the time required to encrypt and decode images with varying formats and dimensions. This indicates that as an image's size increases, so does the processing time.

6 Conclusion

Ransomware assaults make data protection crucial, necessitating the use of reliable technologies like AES. Utilizing four threads for encryption and decryption, this study uses AES to encode and decode multimedia data. Furthermore, this method has examined time utilizing both sequential and parallel processing techniques on a range of formats and sizes of photos. Python is used because it is an object-oriented, user-friendly programming language. The proposed method minimizes the encryption time and guarantees data security over the network, which makes it appropriate for social apps such as WhatsApp.

References

[1] Liang, Q. Qin, C. Zhou, and S. Xu, ‘‘Color image encryption algorithm based on four-dimensional multi-stable
hyper chaotic system and DNA strand displacement,’’ J. Electron. Eng. Technol., vol. 18, no. 1, pp. 539–559,
Jan. 2022, doi: 10.1007/S42835-022-01157-5.

[2] V. Vanitha and D. Akila, ‘‘Bio-medical image encryption using the modified chaotic image encryption method,’’ in Artificial Intelligence on Medical Data (Lecture Notes in Computational Vision and Biomechanics), vol. 37. Singapore: Springer, 2023, pp. 231–241, doi: 10.1007/978-981-19- 0151-5_20.

[3] C. Zhang and B. Du, ‘‘A fast piecewise image encryption scheme combining NC1DNSM and P-box,’’ Integration, vol. 88, pp. 328–342, Jan. 2023, doi: 10.1016/J.VLSI.2022.10.003.

[4] Q. Lai, G. Hu, U. Erkan, and A. Toktas, ‘‘High-efficiency medical image encryption method based on 2D logistic-Gaussian hyperchaotic map,’’ Appl. Math. Comput., vol. 442, Apr. 2023, Art. no. 127738, doi: 10.1016/J.AMC.2022.127738.

[5] L. Liu and J. Wang, ‘‘A cluster of 1D quadratic chaotic map and its applications in image encryption,’’ Math. Comput. Simul., vol. 204, pp. 89–114, Feb. 2023, doi: 10.1016/J.MATCOM.2022.07.030.

[6] W. Song, C. Fu, Y. Zheng, M. Tie, J. Liu, and J. Chen, ‘‘A parallel image encryption algorithm using intra bitplane scrambling,’’ Math. Comput. Simul., vol. 204, pp. 71–88, Feb. 2023, doi: 10.1016/J.MATCOM.2022.07.029.

[7] U. Erkan, A. Toktas, and Q. Lai, ‘‘2D hyperchaotic system based on Schaffer function for image encryption,’’ Exp. Syst. Appl., vol. 213, Mar. 2023, Art. no. 119076, doi: 10.1016/J.ESWA.2022.119076.

[8] Q. Lai, G. Hu, U. Erkan, and A. Toktas, ‘‘A novel pixel-split image encryption scheme based on 2D Salomon map,’’ Exp. Syst. Appl., vol. 213, Mar. 2023, Art. no. 118845, doi: 10.1016/J.ESWA.2022.118845.

[9] O. S. Faragallah, A. I. Sallam, and H. S. El-sayed, ‘‘Visual protection using RC5 selective encryption in telemedicine,’’ Intell. Autom. Soft Comput., vol. 31, no. 1, pp. 177–190, 2022, doi: 10.32604/IASC.2022.019348.

[10] M. Abomhara, O. Zakaria, O. O. Khalifa, A. A. Zaidan, and B. B. Zaidan, ‘‘Enhancing selective encryption for H.264/AVC using advanced encryption standard,’’ 2022, arXiv:2201.03391.