Assessing the Effectiveness of Java Encryption Methods for Image, Video, Audio, and Document Data Types

[1]Auyo SG, [2] Kamal SW, [3]Yushau A, [4] Ibrahim AA
[1] and [3]Department of Computer Science
Jigawa State Polytechnic Dutse
[2] Department of Computer Science
National Open University of Nigeria
[4] Department of Information Technology
Federal University Dutse
salihuauyo@gmail.com

DOI: 10.56201/ijcsmt.v9.no5.2023.pg12.18

Abstract

Confidential data is concealed through encryption, which turns it into an unintelligible format. This study aims to evaluate the efficiency of encryption techniques for Image, Video, Audio, and Document Data Types in Java. This study provides an efficiency comparison benchmarking evaluation to analyze three encryption algorithms i.e. Blowfish, Rivest Cipher2, and Triple Data Encryption Standard to choose the best in terms of size and time efficiency. By evaluating the time/size required by the algorithms to encrypt and decrypt the different data types in different data block sizes using an electronic codebook (ECB), benchmarking has been done using Java. In this study, Rivest Cipher2 (RC2) appears to be the best encryption/decryption algorithm in JAVA programming, outperforming the two most popular encryption algorithms, Blowfish, and Triple Data Encryption Standard (3DES), in benchmarking comparisons at various settings for each algorithm and different size of data blocks. The key factor that distinguishes RC2 as the best algorithm among the two is that it maintains lowest time efficiency followed by blowfish second 3DES with a slight difference and a difference of fewer milliseconds. 3DES is not a significantly worse algorithm, but it takes longer to complete than the other algorithms. Likewise all the three contender's algorithms RC2, Blowfish and 3DES are very good in terms of size efficiency evaluation having maintaining average size.

Keywords: *Blowfish, Graphical User Interface, Decrypt, Encrypt, Rivest Cipher, Triple Data Encryption*

Introduction

While information technology advancements improve our comfort and efficiency, they also present new information security issues Mehmood (2021), increased access to information services will only be possible with secure data transfer (Hahn et al. 2019).

The safety of private documents is guaranteed regardless of whether the machine is accessible to the public Katagiri & Min (2019), and Kashan (2020) suggest that file ciphering using the user's encryption key may be permitted for each user (or a collection of users). This ensures that other users cannot successfully decode and access the plain text because they do not have the encryption key Osamor & Edosomwan (2021). The backup file has additionally been encrypted to produce ciphertext to lessen the risk of data loss brought on by the loss or destruction of the restoration medium (Seth et al. 2022).

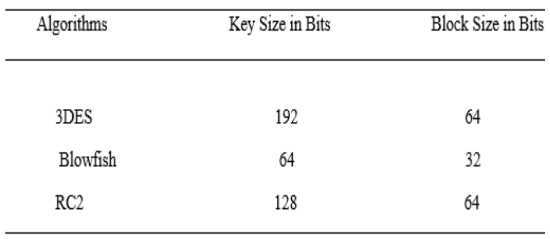
Methodology

This study used benchmarking techniques in evaluating the efficiency of encryption techniques for image, video, audio, and document data types in Java.

Benchmarking Experiment Settings

As demonstrated in Table 1 below, this benchmarking benchmarked the performance of RC2, Blowfish, and 3DES using the supplied classes in the JAVA environment and includes key size and block size for each method.

Table 1 Algorithms Key Sizes



Algorithms	Key Size in Bits	Block Size in Bits
3DES	192	64
Blowfish	64	32
RC2	128	64

Encryption Technique Parameters

An Intel Pentium CPU 3825U 64-bit processor and 4GB of RAM were used to conduct the testing. The benchmarking software was created using the JAVA default settings. To ensure that the results are reliable and that they can be used to evaluate alternative strategies, the tests were run many times. To give users flexibility and control over the encryption process and to enable them to balance security and performance according to their unique demands, encryption approaches employ parameters.

Benchmarking Technique Procedure

The efficiency of the various techniques was assessed using the time required to encrypt and decrypt data blocks of different sizes (50kb, 100kb, 150kb to 200kb) utilizing different sizes of data blocks. Each implementation was done with care to guarantee fair and accurate results. The benchmarking program receives three parameters: Data Block Size, Cipher Mode, and Algorithm Parameters. After a satisfying completion, the facts that were formed, encoded, and decrypted are shown. Character images appear to be the most commonly used kinds. To guarantee that all data is handled properly following a successful encryption and decryption process, the created data (the unique data blocks) are compared to the decryption data block that ended during the process.

Modes of the Data Type Operation

The upload data path type of algorithm selection, the section where the time is taken and the file size of each data type are all contained on separate tabs for the data kinds of text, images, audio, and video. The following criteria are used to assess the data kinds of image, audio, video, and document. The Electronic Code Book (ECB) block cipher has a simple operation and is frequently used with symmetric key encryption. It offers a simple method for processing a list of message blocks that have been arranged in order. Multiple blocks are included in the provided plaintext.

Block Cipher

To analyze the effectiveness of four different data kinds, this study adopted the block cipher mode of operations. Depending on the kind of input, encryption techniques are separated into two categories: block ciphers and stream ciphers. A block cipher is a type of encryption that uses a fixed-size input to create a ciphertext that is also bits in size. The input can be further divided if it is larger than bits. For a variety of uses and applications, a block cipher can operate in a variety of ways (Mehmood 2021).

A fundamental cryptographic blockchain called Electronic Codebook Mode (ECB) is depicted in Figure 1 below, where data blocks are immediately encrypted to produce cryptographic blocks.

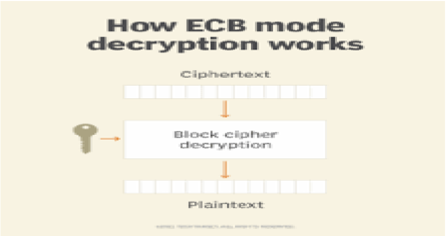


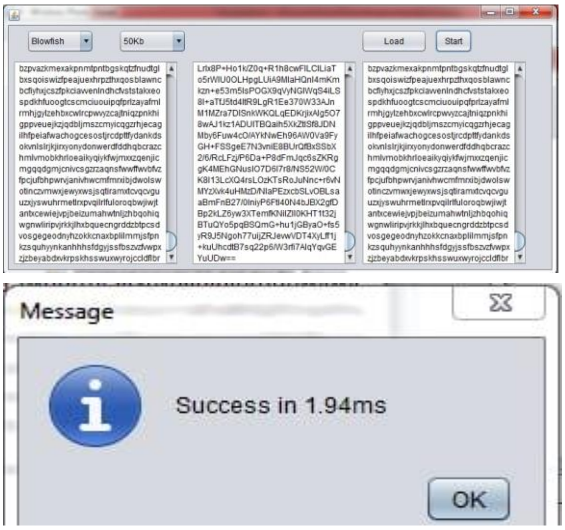
Figure 1: Block Cipher (ECB) Mode

Experimental Setup

User systems engage with interface designers and models repeatedly during the process of user interface design (Seth et al. 2022). The GUI for the benchmarking analysis is shown in Figures 2 and 3.

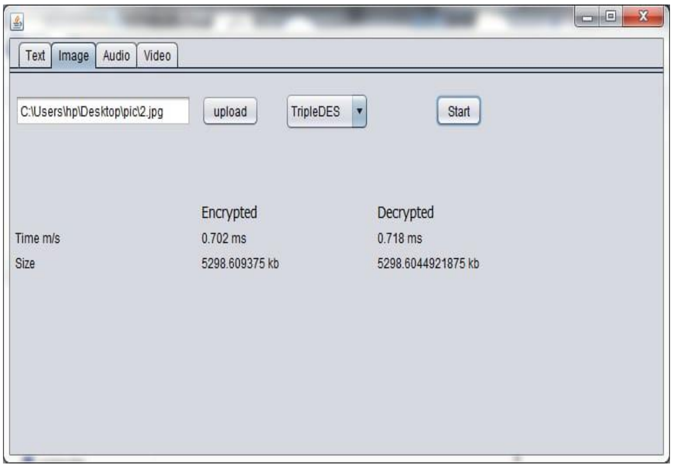
The amount of time required to encrypt and decode data blocks of different sizes (50kb, 100kb, and 150kb to 200kb) was used to evaluate the effectiveness of the various algorithms. To ensure fair and accurate outcomes, each implementation was carefully carried out. Data Block Size, Cipher Mode, and Algorithm Parameters are the three parameters sent to the benchmarking application. Following successful completion displays the data that was generated, encrypted, and decrypted. The majority of the used characters seem to be character photos.

When the user begins the encryption process, Figure 2 below will open up and show the created text on the left, the encrypted text in the middle, and the decoded text on the right.



Success in 1.94ms

Figure 2: Java GUI of Benchmarking Technique



C:\Users\hp\Desktop\pic\2.jpg

TripleDES

	Encrypted	Decrypted
Time m/s	0.702 ms	0.718 ms
Size	5298.609375 kb	5298.6044921875 kb

Figure 3: Triple DES Image selection

The JAVA benchmarking approach will display a successful message following a successful encryption or decryption procedure, as seen in Figure 4 below, where blowfish in 50 kilobytes encrypts and decrypts in 1.94 milliseconds. A similar process applies to 100kb, 150kb, and 200kb for each algorithm.

Image, Audio, and Video Result

The application analysis software launches with four tabs that include text, image, video, and audio. Clicking the image tab in the panel should display an upload button that will enable the upload of images. Then, choose an algorithm and click start. A similar process applies to the rest of the algorithm and the data types as shown in Figure 3 above.

Table 2 Algorithms Encrypt/Decrypt Time

Sample file	Algorithms	Encr Time in ms	Decr Time in ms
Text	Blowfish	0.29	1.07
Text	RC2	0.27	1.08
Text	3DES	0.5	0.6
Image	Blowfish	1.092	0.187
Image	RC2	0.125	0.062
Image	3DES	0.702	0.718
Audio	Blowfish	0.109	0.062
Audio	RC2	0.062	0.031
Audio	3DES	0.172	0.14
Video	Blowfish	0.125	0.047
Video	RC2	0.078	0.031
Video	3DES	0.484	0.171

The results of running the benchmarking study with various data loads are shown in this section. As shown in Figures 4 to 6 below, the results reveal the impact of changing the data load on every technique for each of the examined algorithms (RC2, 3DES, and Blowfish), as well as the impact of employing the Electronic Code Book as a Cipher Mode (ECB).

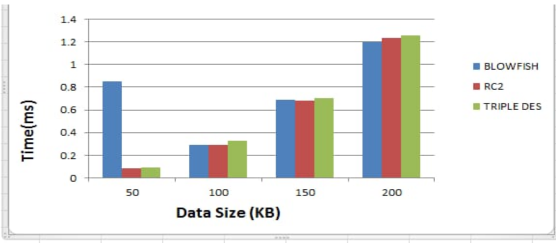


Figure 4: Text, Audio, Video Efficiency Result

This part displays the outcomes of executing the benchmarking analysis with various data loads. The findings indicate the effect of adjusting data's load on every method for each of the compared algorithms (RC2, 3DES, and Blowfish) as shown in figure 4, 5, and 6 below, as well as the effect of using Electronic Code Book as a Cipher Mode (ECB).

Figure 5 below demonstrates how the duration between an encrypted and decrypted file changes depending on the behavior of each method and the kind of file being processed. In the picture file, you can see that the blowfish decryption process required more time than any other method or file format. Triple DES and RC2 offer the lowest decryption times, followed by blowfish. Triple DES is the encryption technique that takes the longest time to complete, followed by blowfish encryption, and RC2 is the fastest of the three. In the same way, RC2 has the fastest encryption and decryption times with a big margin for both RC2 and blowfish, whereas 3DES has a small gap.

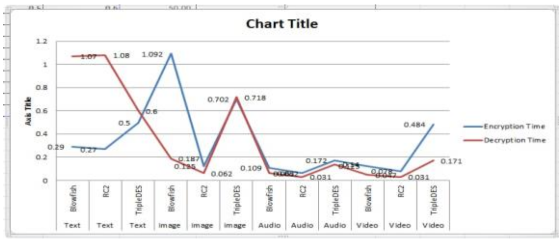


Figure 5: Time Efficiency of Text, Image, Audio, and Video Data Type

Figure 6 below demonstrates that all three techniques employed have the same size for each file utilized in the encryption/decryption process regardless of size reduction or addition.

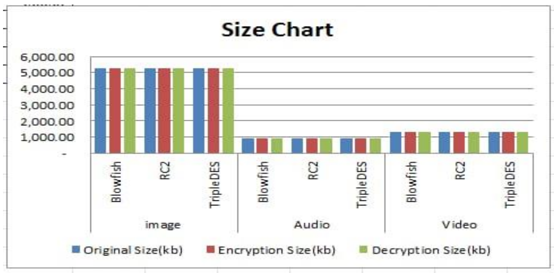


Figure 6: Size Efficiency of Image, Audio, and Video Data Type

The research findings, which are shown in Figures 5 and 6 above, were based on assessing the effectiveness of four data types: picture, audio, video, and text. The results indicate that RC2 is superior to the other ways in terms of time and size efficiency since it retains the lowest time, while all three examined algorithms are excellent in terms of size efficiency, with Blowfish Algorithm coming in second.

The DES appears to work more slowly than the others due to its triple-phase

The other loads on the machine had little effect on these results because each experiment was performed several times and produced results that were almost identical to what was anticipated each time.

The JAVA benchmarking indicates that all the algorithms (RC2, 3DES, and Blowfish) are advised for size efficiency, with RC2 being preferred due to the lowest time maintenance.

Benchmarking findings showed that RC2 outperforms other widely used encryption methods. Since it has no known security issues and takes the least amount of time for data type encryption and decryption, RC2 is a good candidate to be adopted as a standard encryption technology. 3DES didn't perform significantly worse than other algorithms since it needs more capable handling abilities. The processing time and size efficiency gain of using ECB mode were typically small, even for applications that require more secure encryption of relatively large volumes of data.

Conclusion and Recommendation

This study has demonstrated how the fastest method for text, audio, video, and image encryption is produced using the JAVA benchmarking comparison approach. Benchmarking was carried out in Electronic Code Book Mode (ECB) to assess time and size efficiency across multiple data loads (KBs) to ascertain which approach was the fastest. Blowfish and 3DES were contrasted with Rivest Cipher2 (RC2). The relevance of adopting RC2 (symmetric encryption) among discrete users to increase information security, wherever all relevant ideas surrounding information system security have been explored, comes last but not least. Symmetric encryption is recommended, and all three of the competitors' methods are excellent and advised in terms of size efficiency.

References

[1] Hahn, D., Munir, A., & Behzadan, V. (2019). Security and privacy issues in intelligent transportation systems: Classification and challenges. *IEEE Intelligent Transportation Systems Magazine*, *13*(1), 181-196.

[2] Katagiri, A., & Min, E. (2019). The credibility of public and private signals: A document-based approach. *American Political Science Review*, *113*(1), 156-172.

[3] Khashan, O. A. (2020). Secure outsourcing and sharing of cloud data using a user-side encrypted file system. *IEEE Access*, *8*, 210855-210867.

[3] Mehmood, T. (2021). Information technology competencies and fleet management practices lead to *International Journal of Technology, Innovation, and Management (IJTIM)*, *1*(2), 14-41.

[4] Osamor, V. C., & Edosomwan, I. B. (2021). Employing scrambled alpha-numeric randomization and RSA algorithm. *Informatics in Medicine Unlocked*, *25*, 100672.

[5] Seth, B., Dalal, S., Jaglan, V., Le, D. N., Mohan, S., & Srivastava, G. (2022). Integrating encryption techniques for secure data storage in the cloud. *Transactions on Emerging Telecommunications Technologies*, *33*(4), e4108.