

# Image Encryption and Decryption using Enigma Algorithm

Md. Towsif Abir

*Department of Electrical and Electronic  
Engineering  
Bangladesh University of Engineering and  
Technology  
Dhaka, Bangladesh  
towsif.buet.eee14@gmail.com*

Lamiya Rahman

*Department of Electrical and Electronic  
Engineering  
Bangladesh University of Engineering and  
Technology  
Dhaka, Bangladesh  
lrliabd@gmail.com*

Samit Shahnawaz Miftah

*Department of Electrical and Electronic  
Engineering  
Bangladesh University of Engineering and  
Technology  
Dhaka, Bangladesh  
ssmiftah@gmail.com*

Sudipta Sarker

*Department of Electrical and Electronic  
Engineering  
Bangladesh University of Engineering and  
Technology  
Dhaka, Bangladesh  
sudiptasarker.ss@gmail.com*

Md. Ibrahim Al Imran

*Department of Electrical and Electronic  
Engineering  
Bangladesh University of Engineering and  
Technology  
Dhaka, Bangladesh  
ibrahim.buet.eee@gmail.com*

Md. Shafiqul Islam

*Department of Electrical and Electronic  
Engineering  
Bangladesh University of Engineering and  
Technology  
Dhaka, Bangladesh  
shafiqulislam@eee.buet.ac.bd*

**Abstract**—The main objective of this paper is to present a more secured and computationally efficient procedure of encrypting and decrypting images using the Enigma algorithm in comparison to the existing methods. Available literature on image encryptions and descriptions are not highly secured in every case. To achieve more secured image processing for highly advanced technologies, a proposed algorithm can be the process used in Enigma machine for image encryption and decryption. Enigma machine is a piece of spook hardware that was used frequently during the World War II by the Germans. This paper describes the detailed algorithm along with proper demonstration of several essential components present in an Enigma machine that is required for image security. Each pixel in a colorful picture can be represented by RGB (Red, Green, Blue) value. The range of RGB values is 0 to 255 that states the red, green and blue intensity of a particular picture. These RGB values are accessed one by one and changed into another by various steps and hence it is not possible to track the original RGB value. In order to retrieve the original image, the receiver needs to know the setting of the Enigma. To compare the decrypted image with the original one, these two images are subtracted and their results are also discussed in this paper.

**Keywords**— encryption, decryption, algorithm, Enigma, pixel

## I. INTRODUCTION

Security and surveillance systems have advanced so far, yet it has become one of the most popular and challenging issues to work with in this modern world. Day by day, the studies and researches related to security systems have been sprouting swiftly. Encryption and decryption play a vital part in this security purposes. Encryption is a system of mathematical interpretation that encodes a series of data (such as images in this paper) [1] and decryption is the process of decoding the information from the encrypted data for the intended receivers [2]. At present, encryption and decryption are widely

used and have various applications in military communication, security networks, medical imaging etc. to protect privacy and identity of users [3]. In this world of modern technology, these techniques are necessary to send data via the internet so that hackers may not be able to decode them [4].

In the past decade, several studies have been conducted on the encryption and decryption of data. Various papers have discussed the methods of encrypting both text and images [5]. Many have been working on new methods to encrypt and decrypt images while others have been trying to improve the old technologies [6]. Till now several algorithms and studies have been done. One of the methods is chaos method to encrypt images where the grey scale values are shuffled and combined to confuse the relationship between plain and cipher image. [7]. A popular method used in medical purposes that uses two encryption algorithms namely RC4, AES are Stream cipher, Block cipher algorithm respectively [8]. Another paper presents an approach for a random combination of the permutations for image encryption [9]. An algorithm to hide images is also a process of encryption that uses four gray scale images on a single-color image, and encrypt stego-image (hidden image) by using a key [10].

Amongst many algorithms, the technique of the Enigma machine is used in this paper. For better understanding the methodology and process, this paper explores the encryption and decryption of an image using the algorithm of an Enigma machine. Through the ages, several works have been done on encrypting and decrypting images [11] but till now the Enigma algorithm has been barely used in any of those works.

Enigma machines were developed in the early 20<sup>th</sup> century, that uses a series of electro-mechanical rotor in order to protect diplomatic and military communication [12].

The German military models were the most complex, which is used to encrypt messages. This model consists of a keyboard, three rotors, plug-board and a reflector. Straddling the border between mechanical and electrical, this Enigma machine almost looked like an oversized typewriter[13]. Initially, a letter was given as input into the keyboard. Then it passes through all three rotors, bounces off a reflector at the end, and passes back through all three rotors in the other direction. In each step, the letter changes to another. Even though the enigma machine has shown effective outcomes but yet the messages were able to be decoded as the letter never become themselves and many messages ended with similar phases [14].

In this paper, different RGB values of the pixels are taken into account and changed to another RGB value in such a way that whole image cannot be identified anymore and hence encryption is done. To decrypt the image, the setting of the machine has to be known.

## II. METHODOLOGY

Initially few settings in the Enigma have been installed during the World War II. After the encryption of the message, it can only be decrypted if the settings of the machine are known. The settings mainly depend on three major parts-rotors, reflector and plug-board. The proposed method explained in this paper consists of three rotors, one reflector and one plug-board. In this paper, the RGB values at first pass through the plug-board, then consecutively to the first, second and third rotor, and finally to the reflector.

### A. Plug-board, Rotor and Reflector

The plugboard swaps three to six pairs of pixel values before and after a pixel passes through the rest of the program. It can also be programmed in such a way that it swaps blocks of pixels instead, each block containing a number of pixel values (say 20, 30 or 40) based on the level of security required. The operator can configure the plugboard by programming in MATLAB. The RGB values of a pixel are always in the range of 0-255, but MATLAB starts indexing from 1. So, all the parts have been modelled using the range 1-256, mapped the pixel values with 1 incremented, and decremented by 1 at the output. First, the plug-board is modelled by a certain matrix and swapped the desired element pairs (or block pairs) given by the plug-board settings into another matrix.

Next, the rotor positions have to be mentioned by the user. There are three rotors used in this mechanism. These rotors are unique, modeled by three matrices having 256 values in random. These rotors are arranged in different position in the slots and one rotor can be used more than once. So, it's harder to predict which rotor is placed in a certain position. The terminals of the rotors are arranged in circular mode. Each terminal carrier is numbered as the pixels from 0 to 255. Each time a pixel value is passed, the right wheel moves on one of its 256 places. Once during 256 moves, at

the turnover position on right rotor, the middle rotor also moves one place. If the middle rotor reaches turnover position, the last rotor moves as well.

Finally, the reflector is modelled by a matrix which will swap the first number with the last, second with the second last and so on, that is, it gives the complementary number of a certain pixel. Let's take an example, pixel "1" becomes "256" and "2" becomes "255" and this goes on.

### B. Encryption and Decryption

The encryption of each pixel is executed one by one. As shown in the Fig. 1,

**Step-1:** Let's say the first red pixel ("red" value of the pixel on row 1, column 1 of the image) Red-51, used here is passed through the plug-board. It swaps to another value, such as Red-61 here, that has been mentioned initially.

**Step-2:** Next the position of the rotor has to be mentioned by the user and which rotors have been used is also known by the user. Thus, after passing through the rotor 1, the pixel value has been changed into Red-133 which is the value of the first rotor in the 61<sup>st</sup> index.

**Step-3:** Similarly, after passing through rotor 2 and rotor 3, the pixel value goes to the reflector. The reflector complements the pixel value and hence finally it turns to Red-180.

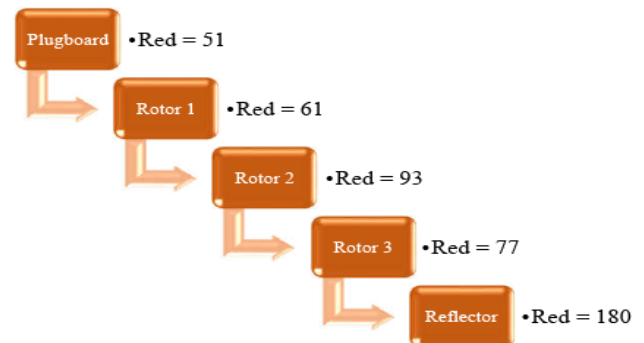


Figure 1: Color Value Changing with each Step Before being Reflected

**Step-4:** Again, this process is reversed as shown in the figure 2. It passes through rotor-3 then rotor 2 and finally rotor 1 to the plug-board and a totally different pixel value is achieved at the end of this process.

Here, an example is explained in terms of a red pixel only. Similar process is also done for the green and blue pixels and all the original pixels have now turned to different ones, distorting the whole picture.

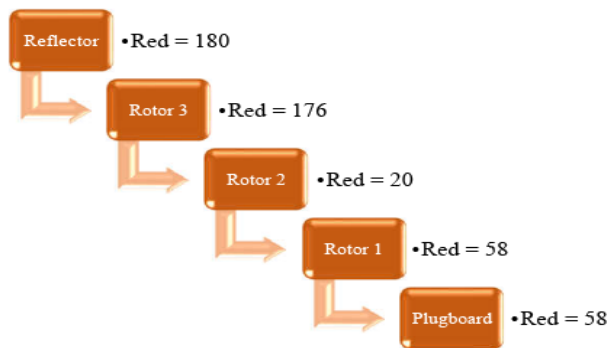


Figure 2: Color Value Changing with Each Step After Being Reflected

The Enigma used a self-reciprocal method and so is this implemented method. There is no difference between the encryption and the decryption process.

For decryption process every setting of the machine that had been mentioned by the user earlier has to be known otherwise, decryption is not possible here.

### III. RESULT

Windows 10 -64bit Computer with specifications as follows:

Computer specs:

Processor : Intel Core i7 7700, 3.6GHz, 64bit

RAM : 12GB DDR4

Graphics Card : Nvidia GTX 1060 3GB

With default settings, the encryption and decryption time for different resolutions of images are as follows in the aforementioned processor:

Duration vs Pixel Graph

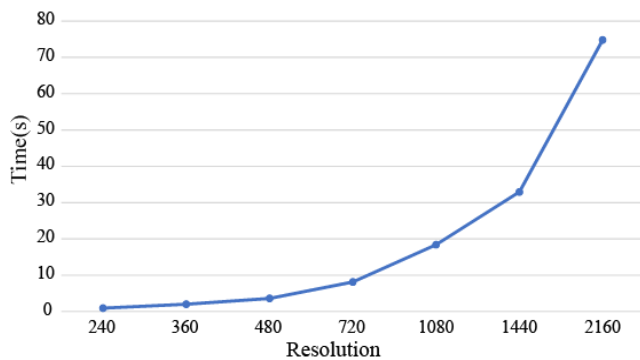


Figure 3: Processing Time vs Resolution Graph

Images of 144p, 240p, 360p, 720p, 1080p, 1440p and 2160p images of aspect ratio 16:9 were taken to get the data. The following graph shows that the relation between time and resolution and it can be concluded that this method is time efficient.

Using our algorithm, an image was encrypted and the results are found as shown below:

Original Image	Encrypted Image

Figure 4: Encryption, (1) Original Image, (2) Encrypted Image

To recover encrypted image, the correct encryption combinations must be known and using the same combination encrypting again will give back the original image. Otherwise, another white noise image will be found as shown as follows:

Decrypted Image (Wrong Combination)	Decrypted Image (Correct Combination)

Figure 5: Decryption, (1) With Wrong Combination (2) With Correct Combination

If compared to its original image placing the decrypted image over the original image and subtracting it gives a pure black image, which implies the minuend and the subtrahend are exactly the same as shown below:



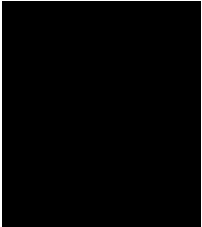


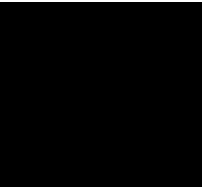


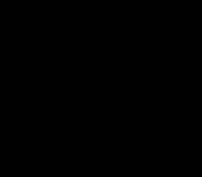


Original Image	Decrypted Image	Difference
		
		
		

Figure 6: Layer by Layer Subtraction of Original image and Decrypted Image

The algorithm is designed in such a way that it works 100% of the time. Around 200 images were tested and it gave the same result 100% of the time.

A plugboard settings has been added to our algorithm so that encryption quality can be controlled. In this demonstration, 6 swapping pair configurations has been added. With this configuration, images can contain minimal to severe noise. Increasing number of swapping pairs will allow a greater variety of control over encryption.

Using pairs of pixel value blocks (each block containing a number of values in serial, say 20 values) to swap instead of pairs of pixels, decryption with wrong plugboard combination gives a resulting image containing such noise as shown below:

Original Image	Decrypted Image (Wrong combination)
	





	
	

Figure 7: Block of pixel value (containing 20 values) swapping Plugboard Setting Decryption

#### IV. Conclusion

Considering the time taken for the encryption and decryption process, this method is fast enough to use for various purpose. Also, based on the requirement of the level of security and ease of use, various encryption parameters can be flexed. The proposed method is not only secured but also maintains integrity. This algorithm is highly suitable for the digitalized system because of the image processing performed efficiently. For higher security purposes, totally unique rotors can be installed other than the default rotors. For day-to-day use, the plugboard option can be turned off which will be easier for users. For this reason, the same user can change the combination of the algorithm and send it to different receivers that can only be accessed by a specific combination which similar to a message with a pin code. Further modifications of this algorithm can be used on file types other than just image files.

#### References

- [1] "Encryption 101: What It Is, How It Works, and Why We Need It," Security News - Trend Micro IN, [Online]. Available: <https://www.trendmicro.com/vinfo/in/security/news/online-privacy/encryption-101-what-it-is-how-it-works>. [Accessed 25 October 2018].
- [2] "Decryption - What is Decryption ? Decryption meaning," Decryption definition - The Economic Times, [Online]. Available: <https://economictimes.indiatimes.com/definition/decryption>. [Accessed 2018 November].
- [3] P. a. S. K. Singh, "Image encryption and decryption using blowfish algorithm in Matlab," *International Journal of Scientific & Engineering Research*, vol. 4, pp. 150--154, 2013.
- [4] "Why do we need encryption? And how does it work?," Beaming, Aug 2018. [Online]. Available: <https://www.beaming.co.uk/support/information-security/why-do-we-need-encryption/>. [Accessed 25 October 2018].
- [5] K. D. a. B. S. Patel, "Image encryption using different techniques: A review," *International Journal of Emerging Technology and Advanced Engineering*, vol. 1, pp. 30--34, 2011.
- [6] S. a. G. C. a. S. J. T. Liu, "A review of optical image encryption techniques," *Optics & Laser Technology*, vol. 57, pp. 327-342, 2014.
- [7] Z.-H. a. H. F. a. G. W. Guan, "Chaos-based image encryption algorithm," *Physics Letters A*, Vols. 1-3, pp. 153--157, 2005.

- [8] G. S. a. K. Amudha, "Medical image integrity control using joint encryption and watermarking techniques," *(2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCCE))*, pp. 1-5, 2014.
- [9] A. a. R. Y. S. a. P. S. a. o. Mitra, "A new image encryption approach using combinational permutation techniques," *International Journal of Computer Science*, vol. 1, pp. 127--131, 2006.
- [10] P. V. R. a. G. N. R. a. P. R. Krishna, "Image encryption after hiding (IEAH) technique for color images," *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, pp. 1202-1207, 2016.
- [11] M. a. A. A. a. G. A. Kumar, "A review on various digital image encryption techniques and security criteria," *International Journal of Computer Applications*, vol. 96, 2014.
- [12] "Enigma machine," Wikimedia Foundation, [Online]. Available: [https://en.wikipedia.org/wiki/Enigma\\_machine](https://en.wikipedia.org/wiki/Enigma_machine), journal={Wikipedia}. [Accessed 25 October 2018].
- [13] A. Hern, "The Guardian," Guardian News and Media, Nov 2014. [Online]. Available: <https://www.theguardian.com/technology/2014/nov/14/how-did-enigma-machine-work-imitation-game>. [Accessed 25 October 2018].
- [14] "What Was the Flaw in the Enigma Machine?," Mental Floss, Apr 2017. [Online]. Available: <http://mentalfloss.com/article/94486/what-was-flaw-enigma-machine>. [Accessed 25 October 2018].