International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 09 | Sep 2019 www.irjet.net p-ISSN: 2395-0072

ENHANCED IMAGE ENCRYPTION SYSTEM USING BLOWFISH AND RANDOMIZATION METHODS

Vincent, Tamaramiebi D[1], Njoku, Celestine N[2], Zuokemefa, Enebraye P[3]

[1]*Lecturer, Department of Computer Science, Isaac Jasper Boro College of Education, Sagbama, Nigeria*
[2]*Senior Lecturer, Department of Computer Science, Federal University of Technology, Owerri, Nigeria*
[3]*Lecturer, Department of Computer Science, Isaac Jasper Boro College of Education, Sagbama, Nigeria*

Abstract - *This research work implements an enhanced image encryption system using blowfish and randomization methods. Images contain sensitive information in compact format and needs to reserve it confidentiality from unauthorized access. Increase in computational speed of modern device and image hacking has ringed the bell for creation and modification of existing image cryptography algorithm and techniques to meet current image security challenges. Hence this work applied image hybridization and separation techniques to provide first level protection for the plain image and blowfish method provided second level protection for the plain image. The system was implemented in Microsoft Windows platform using java programming language, java media framework (JMF) and java image processing (JImage_) to secure joint photographic group (.jpg) image format. The study used scientific/mathematical programming, experimental testing and data visualization methodology. Time complexity and memory size were used to evaluate the system performance. The results shows that's encryption and decryption time depends on the image size, no pixel value was lost during encryption and decryption of images, the system require less encryption and decryption time and also less memory for execution. This system can be applied in Military weapon research and image communication, Agricultural crop research and disease control, Geographic information system, Biological research, patented educational artworks.*

Key Words: Blowfish, Cryptography, Hybridization, Image, Inverse function, Randomization, Separation.

1. INTRODUCTION

In this modern technological age, data and information have become key factor of human existence because various works of life are in dare need of data and information. Typically, data and information are stored in different electronic storage medias such as blue ray, smart card, flash drives (USB drives), memory cards, compact disks (CD), digital Video disks (DVD), hard disks, etc. Data and information stored in these various devices could be files, audio, images, text, animations, documents, etc. To ensure the availability of digital data and information to the world, these digital components need to be transmitted over short and long distances through the internet for various usages and applications. [1] Defines internet as an international network that interconnects computer networks using standard internet protocols such as internet Protocol Suite (TCP/IP) to transmit data and information across the globe to billions of internet users.

The internet is a connection of local, metropolitan to global networks such as public, private, business, government, and academic networks that are joined together by wireless, optical fibres and electronic networking technologies. The use of internet growth geometrically which created the need for internet users to deliberately generate techniques and methods to secure sensitive data and information to deny unfriendly users from getting unauthorized access. The accelerated use of internet by government, academic, health, science, engineering, agriculture, transport, business activities and applications requires a greater demand for quality data and information service. Based on the increasing use of the internet to meet the world's day – to - day data and information activity needs, there exists the urgency to provide quality data and information service by ensuring standard protocol control measures. Hence data and information security is needful and important when transmitted over the internet.

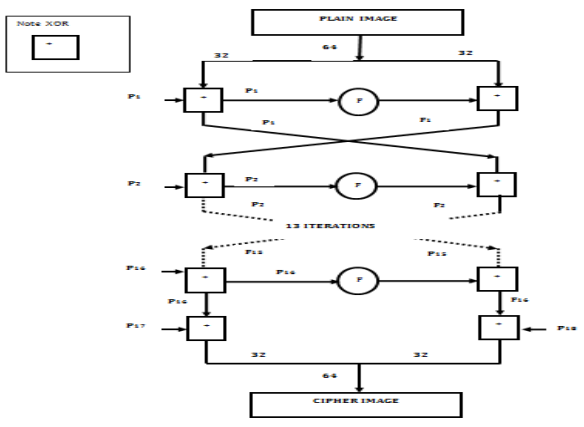
During data transmission, there are chances of these highly confidential, important data could fall into wrong hands such as hackers, which could lead to dangerous situations such as loss of data (bank information, pin numbers, passwords), shut down of emails, websites, servers, telecommunication system etc. However providing data and information security is a complex and broad scope of study. Image data has become one of the most frequently shared and stored data in the world at different end – to - end communication channels. An image data may contain highly confidential and valuable information. There are several types of image data presently in use this modern era. They are applicable in medical research, forensic, military, government sector, multimedia, film division, science research, engineering research, etc. In the cause of image data transmission, hackers do attempt to access the data illegally for malicious use. These illegal actions and troubles mostly occur in the transmission process over the internet [2]. To courageously safeguard and protect image data, cryptography algorithms can be applied.

Today many data encryption algorithms exist such as: Data Encryption Standard (DES), Advanced Encryption Standard

(AES), International Data Encryption Algorithm (IDEA), Blowfish Algorithm, etc. This work focussed on Blowfish Algorithm. Blowfish Algorithm is a symmetric (64-bits) block cipher algorithm that was used to supersede Data Encryption Standard (DES) [3]. This algorithm is very effective in securing data and information in this technological jet age. Its advantage lies on the ranging length of variable key from 32-bits to 448-bits with a default 128-bits, making it one of the best algorithms for securing data and information.

1.1 Blowfish Algorithm

This is one of the symmetric encryption algorithms mostly used effectively for data security. It is available to all users because it is unpatented and license free. It key length ranging from 32bits – to – 448bits making it one of most vital and efficient cryptography algorithms. 1993 Bruce Schneier developed Blowfish algorithm free, faster encryption algorithm alternative to existing algorithms such as DES, 3DES, AES, etc. [4]. Both encryption and decryption uses same secret key and plaintext is divided into fixed block length during these processes. [5] Identified that blowfish algorithm is divided into two parts: data encryption and key expansion. Data encryption applies Feistel Network which simply performs specified iterative rounds of the encryption function while key expansion part divided at most 448bits key length into sub-keys totalling 4168bytes. The 64-bits block length is padded into multiples of eight bytes in size [6]. Presented in figure-1 is the structure of blowfish algorithm. Blowfish keeps two sub-key arrays: Eighteen (18) P-array (32-bits each) boxes and four (4) S-boxes (32-bits each, resulting to 256 entries). After string initialization, the first 32bits of the key are exclusively XORed with the first 32bits P-array (P1). This process continues until the specified round iteration is completed by the algorithm. After the last round of iteration, the left (L) and right (R) are swapped and left (L) is XORed as the 17th index of the P-box and finally the right is XORed as the 18th index of the p-box. This new 64-bit output becomes the cipher data. The decryption process is simply performing a reverse order of the encryption process.



Figure–1: Structure of Blowfish Algorithm [7]

1.2 Feistel Network in Blowfish algorithm

This is a design model from which many different block ciphers are derived. Horst Feistel designed the feistel network model used in several block cipher algorithms (4). Figure-2 represents the use of feistel network model in blowfish encryption algorithm. The iterative rounds depend on the system developer desired number to encrypt plaintext. The iterative rounds consist of substitution and permutation operations. The following are the working steps of a Feistel Network:

1. Input block divided into two halves (Right (R) and Left(L))
2. Right half swapped to become new left half.
3. New left half and function F is exclusively ORed (XORed) to give new right half.
4. Note previous rounds can be derived even without function F.

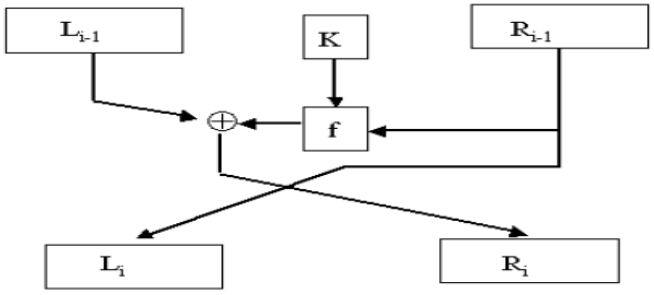
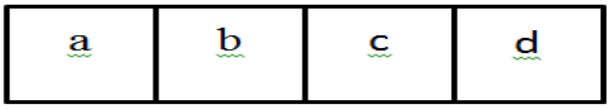


Figure –2: Feistel Network [8]

The function f in Feistel Network is discussed by figure-3. Let the 32-bits input is shared into four segments of 8-bits each as shown by the diagram below.



The function f is generally expressed as:

f (K, R) = ((*sBox*1[*a*]+ *sBox*2[*b*]) ^ *sBox*3[*c*]) + *sBox*4[*d*])

Where sBox1, sBox2, sBox3 and sBox4 are the four S-Boxes and a, b, c, d represents the 32-bits input to the function F, separated into four consecutive discrete segments of 8-bits each, starting from the most significant bit (MSB). Figure 2.15 shows how function F obtains sub-keys for the Feistel Network.

Using the equation 2.3:

A modulo addition is done on sBox1[a] and sBox2[b], the result XORed with sBox3[c]. The XORed result and sBox4[d] are added together to obtain a sub-key.

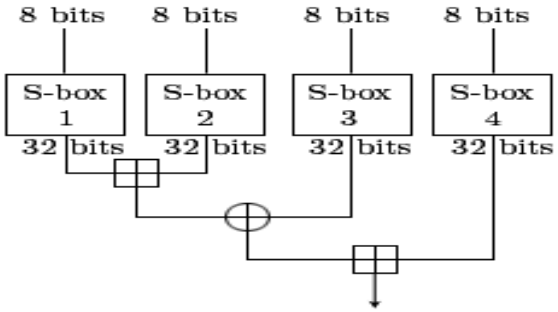


Figure-3: Function f in Feistel Network [7]

1.3 Generating Blowfish Algorithm Sub-keys

[5] Noted that the encryption and decryption are same in blowfish except that during decryption the keys are used in reverse order. They highlighted the following identical steps of blowfish algorithm:

1. Initialization of P-array and four S-boxes with fixed string input.
2. Repeated XOR operation is performed from the first P1 (32-bits) until a possible P14, the cycle is repeated with the key bits.
3. Blowfish algorithm uses Sub-keys described by Step 1 and 2 to encrypt all zero-string.
4. The result of step 3 is used replace P1 and P2.
5. The result of step (3) is encrypted using the blowfish algorithm with the new modified sub-keys.
6. Replace the result of P3 and P4 with the result of (5)

The process continuously replaces P-arrays and S-boxes until all entries are complete, implying that the result also changes continuously. For example if the iterative rounds are sixteen (16). A total of 521 iterations required to generate blowfish algorithm sub-keys.

2. RELATED WORKS

In this modern era, multimedia applications are fast growing in geometric progression and digital data security has become vital issue in image transmission and image storage. Cryptography is one of the several mechanisms to security digital images [9]. Image cryptography transform a plain image into cipher image that is difficult to identify and decipher, ensuring their confidentiality between users. Several image applications require special and reliable systems to protect digital images. At present different methods or techniques can be used to obtain image encryption. Image encryption can be achieved with traditional encryption methods (AES, DES, Blowfish, RSA, etc.), non-traditional methods (chaotic and quantum theories) and encryption content (direct and selective encryption). Direct encryption (pixel by pixel) adopts a simple method of changing the original pixel value of the image while selective encryption approach preform bit streaming that is parsed in order to identify the parts subjected to encryption [10].

[10] Highlighted a detailed analysis of open source computer vision (open CV) methods to explore image data structure. It utilise a defined pointer to traverse image data (making various operations done on the image possible) and integrating it with 2dimensional Arnold transformation technique to encrypt image. To obtain decryption inverse operation is done Arnold transformation. Open CV library functions ensure simple and feasible encryption process and lays the foundation for operational update. Arnold transformation is a simple image encryption method but when larger and increased pixel data is used the process become very slow.

[3] Employed blowfish algorithm to encrypt and decrypt both coloured and black and white images of any size saved in the format of TIF, bmp, PNG, JPG, JPEG, etc. and implemented in MATLAB. Although the histogram used in illustrating the encrypted image was less dynamic, it was significantly different to the respective histograms of the original image. The study showed that an attacker must try $28^r + 1$ possible combinations (r denote iterative rounds) to break a blowfish algorithm. The amount of iterative rounds determines the strength of blowfish algorithm. No blowfish security system is yet to record any known weak point, hence it can be consider as one of the existing excellent standard algorithm for image encryption [3].

[9] Applied two techniques called spatial domain and wavelet domain techniques in their attempt to provide image security. The spatial domain technique performs spatial domain operation on image to use a cover image to hide the desire secret image. The wavelet domain technique hides the secret image in the decomposed cover image data. The decomposed image data bits can be retrieved by using other bits to hide the secret image. A two-stage encryption algorithm was used to finalise the encryption process. The first stage decomposed the secret image into two shared images and the second stage embedded the shared image into the cover image. It was observed that the method proposed and applied by [9] improved the security of secret images.

Furthermore, [11] employed a new technique called watermarking (digital image watermarking) technique. The proposed method encrypts and embeds a watermark into another watermark and these combined watermarks are enclosed into the main image. The act of embedding one watermark into another is known as Nested watermarking. By applying nested watermark it increases the security level of the main image due to the use of encryption and decryption techniques, which also enhances the embedding capacity of the watermark. The proposed technique used blowfish encryption and decryption algorithm which is a suitable and efficient technique for hardware implementation. This proposed technique has the following

advantages: (1) the concept of nesting increases embedding capacity of watermark into the main image, (2) that before the embedding watermarks into the main image an Encryption of watermarks is done to provide a better security, and (3) Blowfish encryption helps to make the method robust [11].

A recent work done by [12] in their paper *"Image Encryption Technique using Blowfish Algorithm"* used a chaos-based technique and blowfish algorithm cryptography to provide an image security system that reduces time taken to encrypt/decrypt, increase efficiency and reliable ways of dealing with bulky, difficult and unruly images. The research recommends that multimedia images should be secured with applied techniques. The reason being that images are arranged in blocks when viewed. Every intelligible image is formed by correlating its elements in a given arrangement. They also applied chaotic technique to divide the image into random blocks which are now shuffled within the image. The new transformed image is fed as input to blowfish encryption. The seeds are determined by a secret key plays a key role in building the transformation table and then generate the transformed image with different random block sizes. Hence, the transformation process is act of dividing and replaces of image arrangement process. The image is divided into blocks, each contain specific number of pixels. The transformed blocks are placed into new locations. At decryption level, the blocks are reversed using inverse transformation to retrieve the original image. The aim of this technique was to enhance the image security level by reducing image elements correlation and increase amount of information value.

3. ANALYSIS OF EXISTING SYSTEM

[3] Used MATLAB to implement an image security system using blowfish encryption algorithm. The architecture of the existing system is presented in figure 3.1. It is designed to encrypt 64-bit block plaintext into same cipher text securely and efficiently. The algorithm operations selects from addition, bitwise logic gate (exclusive-OR) and lookup table to reduce data encryption and decryption time. The algorithm developers consciously maintain simplicity of operations and coding without compromising the system. The researchers blended 16 Feistel Network iterative rounds for encryption and decryption. Each iterative round involves left and right 32-bits data modified. Bitwise exclusive-OR gate is implemented on the left 32-bits before being modified by blowfish function (f) or passed to the right 32-bits for the next Feistel Network iteration round. Swapping operation is done immediately after the 16 iteration round and two bitwise exclusive-OR is performed to produce the cipher text. This actually different from permutation functions done by other cryptography algorithms.

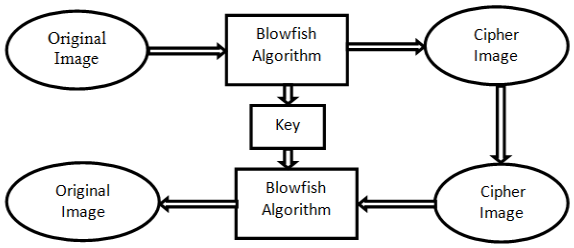


Figure-4: Architecture of the existing Blowfish Encryption system [3]

3.1 Analysis of the Proposed System

This research proposes an image cryptographic system that takes blowfish algorithm (presented in figure-1) as an open-ended design and incorporate randomized hybridization to produce a more secured JPG image pixel-by-pixel cipher blending of the original and randomly generated image to produce inputs to the blowfish algorithm. The output is a cipher image that can be stored, transmitted and decrypted by authorized receiver. The decryption process is to follow the reverse order of blowfish algorithm. The functionalities of the proposed system are specified in phases as follows:

Phase 1: Input Image Generation

1. Generate a random image (.jpg)
2. Get the original image (.jpg) to be encrypted
3. Hybridize original and randomly generated images to form one image (input image)

Phase 2: Image Encryption

1. 16 rounds of Feistel Network iterations
2. Applying Blowfish Algorithm on the input image
3. Generate Encrypted Image (Cipher Image) and store it.

Phase 3: Image Decryption

1. Input Cipher Image
2. Applying inverse of Blowfish Algorithm to decrypt cipher image.
3. Generate decrypted Image (.jpg) to produce the hybridized image.

Phase 4: Generate Original Image

1. Get the decrypted image (i.e. the hybridized image).
2. Get the random image
3. Separate the random image from decrypted image to generate original image.

For system to perform hybridization and separation is important to consider invertible functions.

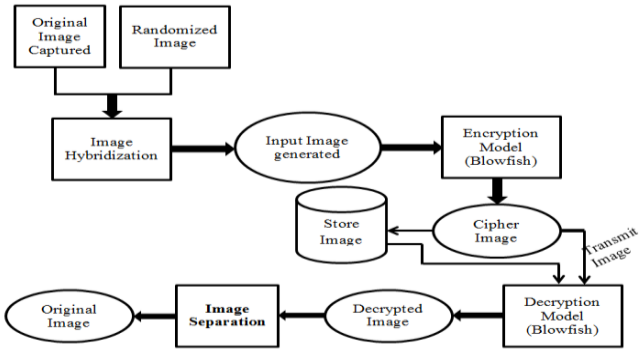


Figure-5: Architecture of the Proposed System Model

3.2 Invertible Functions and Applications to Image Encryption

An inverse function is a function that will undo anything that the original function does. If *f(x)* represents a function, then the notation $f^{-1}(x)$, read *f* is an inverse of *x*, will be used to denote the inverse of *f(x)*[13]. Inverse function can be defined mathematically below:

Let P, Q be sets, and let f: $P \rightarrow Q$. An i*nverse function* for f is a function $g: Q \rightarrow P$ such that

$$f(g(b)) = b \text{ for all } b \in Q \text{ and } g(f(a)) = a \text{ for all } a \in P$$

Note not all function are invertible function hence is vital to determine if a function is invertible. To determine if a function has an inverse function property is necessary to consider Injective (one-to-one) functions. One-to-one function is a function type that where each input value (x) has a peculiar output (y).

Injective functions is a function f from set P to set Q is a function such that each a in P is related to *a different b* in Q.

formally

f:P →Q Injective (one - to - one) function

$f(a) = f(b)$ implies $a = b$, or

f:P →Q

$a \neq b$ implies $f(a_1) \neq f(b)$.

The sketch below illustrates one-to-one mapping of an invertible function.

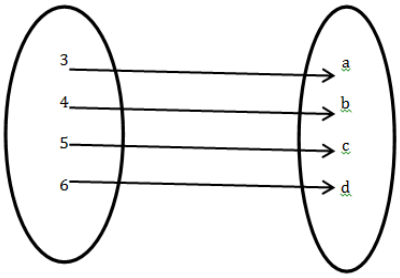


Figure-6: Inverse Function Mapping [13]

Example of an invertible function: The function $f(x) = x + 2$ from the set $P = \{3, 4, 5, 6\}$ to the set $Q = \{5, 6, 7, 8\}$ can be written as follows.

$f(x) = x + 2$: {(3, 5), (4, 6), (5, 7), (6, 8)} the **invertible function** f^{-1} is a function from the set Q to the set P, and can be written as follows.

$f^{-1}(x) = x - 2$: {(5, 3), (6, 4), (7, 5), (8, 6)}

The sketch below shows the mapping relationship between function $f(x)$ and invertible function $f^{-1}(x)$

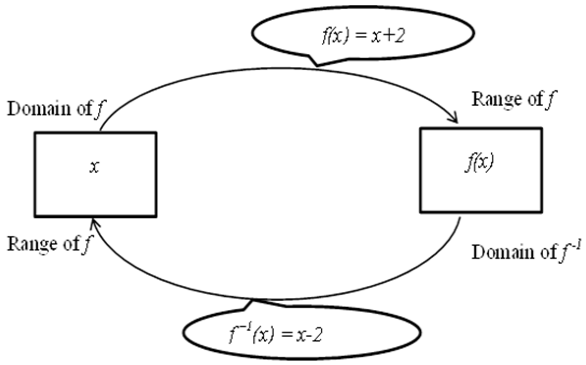


Figure-7: Inverse Function Mapping of f(x) and $f^{-1}(x)$

Applying inverse function to the proposed system

We let P = Original image and Q = Random Image

Were

P and Q contain picture elements or word pixels (red, green, blue) each element being represented with integer values.

Suppose

$f(P, Q, Z) \rightarrow Z = P + Q$

$$P = \begin{bmatrix} 2 & 7 & 6 \\ 5 & 3 & 4 \\ 6 & 2 & 0 \end{bmatrix} \qquad Q = \begin{bmatrix} 5 & 1 & 4 \\ 9 & 7 & 1 \\ 3 & 4 & 6 \end{bmatrix}$$

Let *f(P, Q)* be a matrix function such that *Z = f(P, Q)* is the image hybridization technique where by pixel-by-pixel addition produces a new image. Thus, with the given values of P and Q.

$$Z = \begin{bmatrix} 7 & 8 & 10 \\ 14 & 10 & 5 \\ 9 & 6 & 6 \end{bmatrix}$$

Suppose p_{ij} = [0,1,2,…,255], q_{ij} = [0,1,2,…,255] and z_{ij} = [0,1,2,…,255]

For image hybridization technique

Where

$$z_{ij} = p_{ij} + q_{ij}$$

if ($z_{ij} > 255$) then

$$z_{ij} = z_{ij} - 255$$

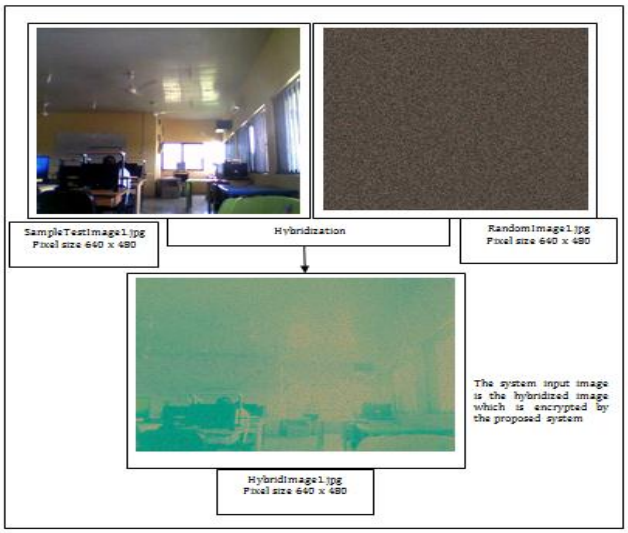


Figure-8: System Input Image Hybridization

The inverse function *f(P, Q, Z)* is denoted by $f^{-1}(Q, Z, P)$ which is a mapping from the matrices Z and Q to matrix *P*.

This can be expressed *as* $f^{-1}(Q, Z, P) \rightarrow P = Z\text{-}Q$

Here the inverse function is a form of image separation technique

$$P = \begin{matrix} 2 & 7 & 6 \\ 5 & 3 & 4 \\ 6 & 2 & 0 \end{matrix}$$

For image separation technique

Where

$$p_{ij} = z_{ij} - q_{ij}$$

if ($p_{ij} < 0$) then

$$p_{ij} = p_{ij} + 255$$

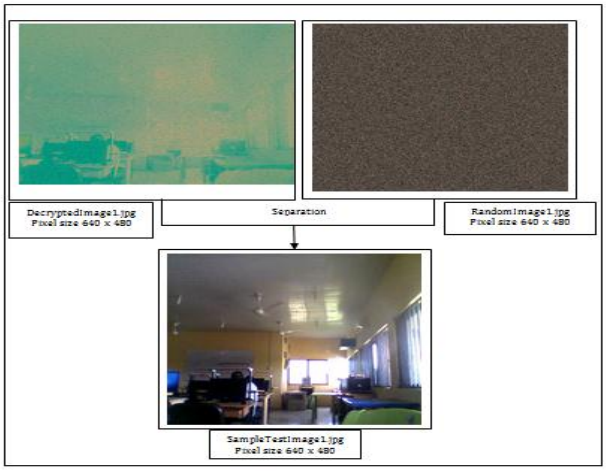


Figure-9: System Output Image Separation

4. ANALYSIS OF RESULTS

The results of the new model were analyzed and compared using performance measurement factors:

i. Time complexity

Tables and charts below illustrate time (speed) complexity performance measure in terms of seconds. Images presented are results of encryption and decryption process of the system with relation to time taken for execution.

Table -1: Comparing encryption and decryption time of system test images.

S/N	File name	Image Size (kb)	Encryption Time (s)	Decryption Time(s)
1	SampleTestImage1.jpg	31.80	0.56	0.59
2	SampleTestImage2.jpg	28.50	0.52	0.50
3	SampleTestImage3.jpg	21.90	0.44	0.41
4	SampleTestImage4.jpg	16.90	0.34	0.32
5	SampleTestImage5.jpg	20.00	0.37	0.35

The chart below compares time taken to encryption and decryption system test images.

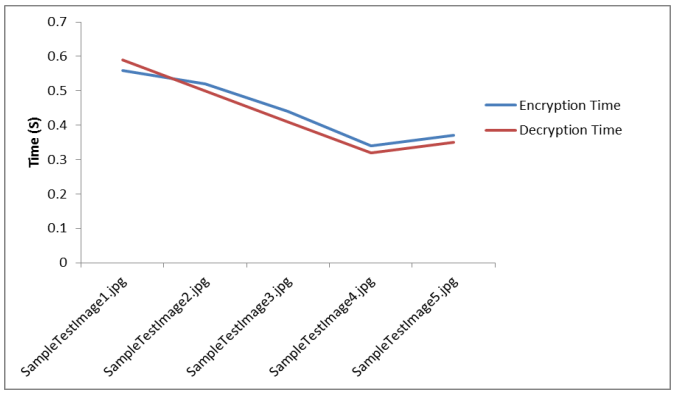


Chart-1: Comparing encryption and decryption time system test images

The test images are hybrided with system generated random images to produce system input images "HybridImage" as shown in figure-8 above.

Table -2: Comparing enhanced and unenhanced encryption time of system input images (HybridImage).

S/N	File name	Image Size (kb)	Enhanced (s)	Unenhanced(s)
1	HybridImage1.jpg	109.00	1.05	1.36
2	HybridImage2.jpg	105.00	0.65	1.27
3	HybridImage3.jpg	88.10	0.63	1.05
4	HybridImage4.jpg	86.40	0.59	1.02
5	HybridImage5.jpg	86.20	0.54	0.57

The chert below compares time taken to encryption system input images by the enhanced system and unhanced system.

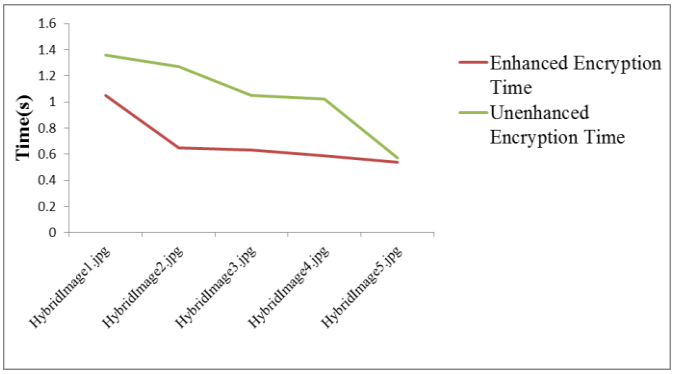


Chart-2: Comparing encryption time difference of input image between enhanced and unenhanced system

Table -3: Comparing enhanced and unenhanced decryption time of system input images.

S/N	File name	Image Size (kb)	Enhanced (s)	Unenhanced(s)
1	HybridImage1.jpg	109.00	1.02	1.23
2	HybridImage2.jpg	105.00	0.68	1.18
3	HybridImage3.jpg	88.10	0.61	1.03
4	HybridImage4.jpg	86.40	0.56	0.68
5	HybridImage5.jpg	86.20	0.51	0.65

The chert below compares time taken to decryption system input images by the enhanced system and unhanced system.

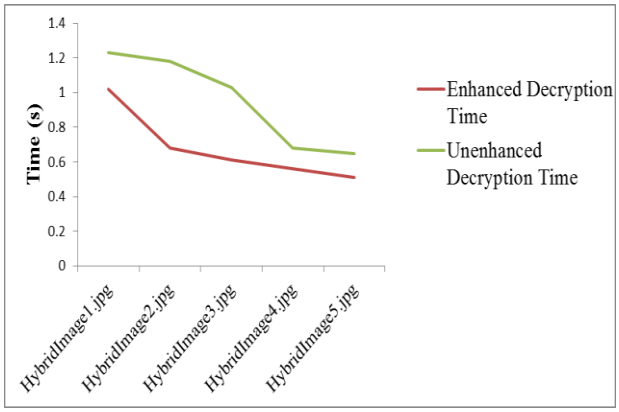


Chart-3: Comparing decryption time difference of input image between enhanced and unenhanced system.

The chart shows that images with smaller memory size require minimal time for encryption and decryption. Chart-1 shows that the enhanced system takes lesser time to decrypt than to encrypt. Chart-2 and chart-3 give a clear picture that the enhanced system takes lesser to encrypt and decryption when compared to unenhanced system.

ii. Memory (Space)

The size of image files is measured before and after encryption to observe the memory consumption. Table4 and table5 shows the different image file sizes of both the encryption and decryption processes of the system in bytes.

Table -4: Comparing Input image system after encryption and decryption.

S/N	File name	Image Size (kb)	Encryption (kb)	Decryption(kb)
1	HybridImage1.jpg	109.00	138	138
2	HybridImage2.jpg	105.00	122	122
3	HybridImage3.jpg	88.10	156	156
4	HybridImage4.jpg	86.40	141	141
5	HybridImage5.jpg	86.20	147	147

Encrypted and decrypted image are equal. Which implies that no pixel value of the images where lost. However the random image most be separated from the decrypted image to get the plain imaged used in the hybridization technique.

Table -5: Comparing separated image with system test image sizes after decryption.

S/N	File name	Image Size (kb)	Test Image (kb)	Separated Image (kb)
1	DecryptedImage1.jpg	138	31.80	31.80
2	DecryptedImage2.jpg	122	28.50	28.50
3	DecryptedImage3.jpg	156	21.90	21.90
4	DecryptedImage4.jpg	141	16.90	16.90
5	DecryptedImage5.jpg	147	20.00	20.00

5. CONCLUSION

This research work enhanced image encryption system using blowfish and randomization methods provided a double protection of the plain image, reduces encryption and decryption time, require minimal memory to execute and prevent pixel value loss during encryption and decryption.

5.1 Recommendations

We recommend the following:

1. Image encryption system using blowfish encryption algorithm should be implemented in hardware.
2. Image encryption system with blowfish algorithm, JMF and JImage_ should be implemented on other image formats (gif, png, tiff, etc.).
3. The number of iteration in blowfish encryption algorithm should be increased to 20 and above.

References

[1] M. Anand and S. Karthikeyan, "Investigating the Efficiency of Blowfish and Rejindael (AES) Algorithms", I.J. Computer Network and Information Security, Mar. 2012,2,22-28, doi:10.5815/ijcnis.2012.02.04, *http://www.mecs-press.org*

[2] J Abdul and M. Jisha, "Guarding Images using a Symmetric key Cryptographic Technique: Blowfish Algorithm", IJEIT vol.3, Issue 2, Aug. 2013, ISSN: 2277-3754

[3] S. Pia and S. Karamjeet, "Image Encryption and Decryption using Blowfish Algorithm in Matlab", IJSER vol. 4, Issue 7, Jul. 2013, ISSN 2229-5518, *http://www.ijser.org*

[4] M. Saikumar and K. Vasanth, "Blowfish Encryption Algorithm for Information Security", Journal of Engineering and Applied Science, ARPN vol.10, No.10, 2015, www.arpnjournals.com

[5] H. Chaitali and K. Sonia, "Implementation of AES and Blowfish Algorithm", IJRET, vol. 3, Issue 3, NCRIET-2014, http://www.ijret.org

[6] N. Valmik and V. Krshirsagar, "Blowfish algorithm", IOSR Journal of Computer Engineering, vol. 16, Issue 2, pp 80-83, Mar-Apri. 2014, e-ISSN: 2278-8727, www.iosrjounals.org

[7] L. Christina and I. Joe, "Optimized Blowfish Encryption Technique", International Innovative Research in Computer and Communication Engineering, vol. 2, Issue 7, 2014, ISO: 3297:2007, www.ijircce.com

[8] W. Edward and T. Achileas, "Blowfish", http://www.ed-wolf.com/wordpress/wp-content/upoads/2011/10/Blowfish_Report.pdf

[9] T. Sudha and B. Gopi, "Novel Spatial and Transform Domain Image Encryption Algorithms", IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) vol. 4, Issue 2, ver. II, 2014, www.iosrjournals.org

[10] P. Ashish, A. Arjun, et al, "Sophisticated Image Encryption using OpenCV", International Journal of advanced Research in Computer Science and Software Engineering, vol.2, Issue1, 2012, www.ijarcsse.com

[11] G. Preeti and M. Neeraj, "A Survey on Image Encryption and Decryption using Blowfish & Watermarking", 3285 IJRITCC vol. 3, Issue 5, 2015, http://www.ijritcc.org

[12] B. Kalidas, S. Yashodha, et al, "Image Encryption Technique using Blowfish Algorithm", IJAFRSE vol. 1, Special Issue, impact factor: 1.036, Science Central value: 26.54, 2015.

[13] C. Sid, "Functions and Inverses", CS 2800: Discrete Structures, Spring, 2015