SECURE PATIENT RECORD STORAGE IN CLOUD USING MODIFIED ELGAMAL AND SAFE ANALYSIS THROUGH HOMOMORPHIC PROPERTY

A.Vikram, Dr. Gopinath Ganapathy

Abstract: Data breach in cloud storage is a major issue that is still prevailing in cloud environment. Patient data is one of the sensitive information that is of interest to adversaries. In recent days, many techniques for securing the data has been evolving and techniques to break those schemes are also evolving. In this paper, an effort has been made to secure patient data – text and scan images, using modified Elgamal for text and permutation preceded blowfish scheme for scan images. Authorized users are granted access control using Role based Access Control (RBAC) mechanism. The modified Elgamal for text and permutation for image adds additional layer of security to the data. It is observed that the modified Elgamal supports homomorphic property. The application of homomorphic property in analysing the encrypted data is also discussed and the comparative results with other existing methodologies are stated.

Key words: Elgamal, RBAC, Permutation, Homomorphic property

I INTRODUCTION

Cloud is a group of computers that are connected together for computing large tasks, voluminous storage and countless services. Cloud provides many services out of which three are predominant – Infrastructure as a Service (IaaS), Software as a Service (SaaS) and Platform as a service (PaaS). IaaS provides storage space for users to store their huge volume of data. When these data is stored in a public cloud, it leads to security compromise of data that is being stored. Data breaches can be addressed by including novel security schemes so that adversary is unable to retrieve the data. Even if the data is somehow retrieved by the adversary it should be of no use or it should consume non polynomial time for decryption. Data is of many forms – text, image, audio, video etc., and this work focuses on securing text and image in public cloud storage. Text is encrypted using a modified Elgamal scheme and image is encrypted using Blowfish algorithm which is preceded by a permutation phase. The authenticity of the user is verified using Role based access control mechanism (RBAC). RBAC exhibits data based on the role of the user. The analysis of data can be done using homomorphism property. There are three types of homomorphisms – Somewhat homomorphic, Partial homomorphic and fully homomorphic. The homomorphic encryption is categorized based on the type of mathematical operation that the encryption algorithm supports. Elgamal is a partial homomorphic algorithm and it supports only multiplicative homomorphic property.

- *A.Vikram School of Computer Science and Engineering, Bharathidasan University, Tiruchirappalli*
- *Gopinath Ganapathy Professor, School of Computer Science and Engineering, Bharathidasan University, Tiruchirappalli*

Farhan Bashir Shaikh in [1] has discussed in detail about the security issues in cloud computing. Sandeep K. Sood [2] has proposed a MAC based encryption for data and has also utilized Secure Socket Layer (SSL) 128 bit encryption and 256 bit encryption. Juels et al. [3] has described a method to ensure data integrity in cloud storage. They have implemented Spot checking and error correcting code in their work. In [4], Pedro Ramos Brandao has discussed the importance of authentication and encryption in cloud computing framework.A review on various image encryption techniques is discussed in [5]. Geetanjali Sinha [6] has done a survey on access control on patient records. A modified Elgamal cryptosystem has been proposed by Prashant Sharma [7]. In [8], Pia singh has shown the encryption and decryption of image using Blowfish. Motivated by the fact that the existing methodologies does not provide an efficient way for secure cloud storage, a novel approach that addresses the cloud storage issues is proposed in this work. The paper is organised as follows, Section 2 describes the entire storage and retrieval process of the proposed work. Section 3 has experimental and results part followed by conclusion in Section 4.

II PROPOSED WORK

The sensitive patient record can be categorized as text data and image data. Different measures should be utilized to handle two different types of data. Also there are two processes – Storage and retrieval in cloud.

A. Storage

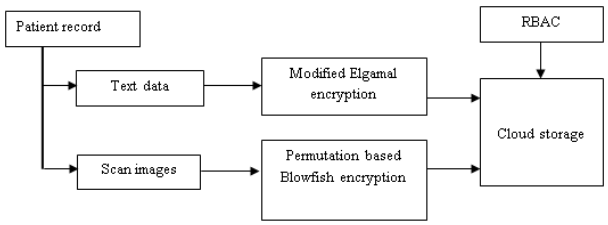


Fig 1: *Storage procedure*

In general, patient records contain both text data and image data and hence demands suitable methodology to handle them. In this work, text data is modified using modified Elgamal. For clarity, traditional Elgamal is first discussed (Section a) followed with modified Elgamal (Section b). Later this is followed by description of permutation based Blowfish encryption (Section c). The last part of storage procedure contains description about Role based Access Control (Section d).

a) Traditional Elgamal

The traditional Elgamal encryption scheme involves the following steps,

- **Key generation**

To generate key, consider a cyclic group Z_p where p is a large prime that is at least 1024 bits (Approximately 309 digits) and let 'g' be its generator. The elements of Z_p include {1,2,…,p-1}.

$$g \in Z_p \bmod p$$

Private key, x: Choose x ϵ {2,3,…,p-2}. Choosing x = 1 will reveal the generator and choosing x = p-1 will result in the value 1.

Public key, y: Compute 'y' as g^x

- **Encryption**

Elgamal encryption has two parts. Let the first part and second part be denoted as C_1 and C_2 respectively. Choose another value called 'k' (k ≠x).

$C_1 = g^k \bmod p$, where k ϵ {2,3,…,p-2}

$C_2 = m * y^k \bmod p$, where 'm' is the plain text

The set (C_1, C_2, x) is sent to the receiver. The value 'k' is unknown to the receiver.

- **Decryption**

The plain text can be retrieved from the cipher text as follows,

m = $C_2 * (C_1^x)^{-1} \bmod p$

Proof:

$C_2 * (C_1^x)^{-1} \bmod p$

$=m * y^k((g^k)^x)^{-1}$ mod p

$=m*(g^x)^k((g^k)^x)^{-1}$ mod p

$=m*g^{xk}g^{-xk} \bmod p$

$=m*g^{xk-xk} \bmod p$

$=m*g^0 \bmod p$

=m mod p

=m

Thus the plain text is derived from cipher text without the knowledge of 'k'.

The strength of Elgamal is based on two hardness problems,

1. Computational Diffie Hellman (CDH) problem – In a cyclic group, given (g, g^x, g^k) where 'g' is a random generator and values x, k are also random, it is hard to compute g^{xk}.
2. Discrete Logarithm Problem (DLP) – Given $y = g^x$, it is hard to find x.

b) Modified Elgamal

Adding to the strength of traditional Elgamal, a contribution is provided in the proposed work to enhance the security of Elgamal. A concept called double exponentiation is introduced. The modified Elgamal scheme based on double exponentiation is described as follows,

- **Key generation**

To generate key, consider a cyclic group Z_p where p is a large prime that is at least 1024 bits (Approximately 309 digits) and let 'g' be its generator. The elements of Z_p include {1,2,…,p-1}.

$$g \in Z_p \bmod p$$

Private key, x: Choose x ϵ {2,3,…,p-2}. Choosing x = 1 will reveal the generator and choosing x = p-1 will result in the value 1.

Public key, y: Compute 'y' as g^x

- **Encryption**

Step 1: Choose random 'k', k ϵ {2,3,…,p-2}and k≠x

Step 2: Choose random 'r', r ϵ {2,3,…,p-2} and r≠(x, k)

Step 3: $C_1 = (g^k)^r \bmod p = g^{kr} \bmod p$, where k ϵ {2,3,…,p-2}

Step 4:

$C_2 = m * (y^k)^r \bmod p = m * y^{kr} \bmod p$

, where 'm' is the plain text

The set (C_1, C_2, x) is sent to the receiver. The value 'k' is unknown to the receiver.

c) Permutation Based Blowfish Encryption

The scan image is first converted into pixels. Next, the image is spitted with arbitrary window size. Using random generator, permutation sequence can be obtained and it is stored in an index. With the permutation sequence, the image blocks are scrambled and this scrambled image will be given as input for encryption with Blowfish scheme. The encrypted image can now be stored in the cloud. This procedure is shown in Figure 2.

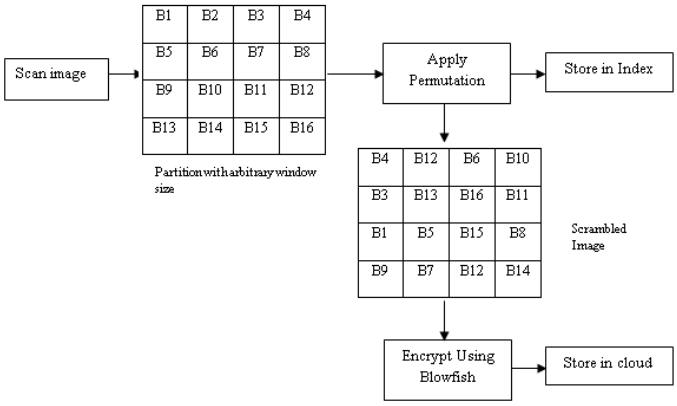


Fig 2 *Permutation based Blowfish Encryption*

Including permutation on image blocks adds additional layer of security. The permutation order is stored in an index for future reference while performing retrieval.

d) Role Based Access Control

Role based access control mechanism is employed to restrict the access of unauthorized user. Access and readability is defined based on the role of the user. This mechanism ensures that only authorized users have access to the sensitive patient records.

B. Retrieval

The retrieval procedure is shown in Figure 3.

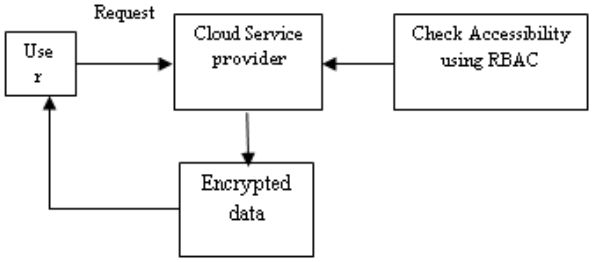


Fig 3 *Retrieval Procedure*

The user requests the cloud service provider with his credentials. The cloud verifies the authenticity of the user with the help of Role based access control mechanism. Once it has been verified that the proper user has sent request, the cloud service provider will return the encrypted data ie., text and image in encrypted format.

a) Text Decryption

The plain text can be retrieved from the cipher text as follows,

m = $C_2 * (C_1^x)^{-1} \bmod p$

Proof:

$C_2 * (C_1^x)^{-1} \bmod p$

=$m * y^{kr}((g^{kr})^x)^{-1}$ mod p

=m*$(g^x)^{kr}((g^{kr})^x)^{-1}$ mod p

=m*$g^{xkr} g^{-xkr} \bmod p$

=m*$g^{xkr-xkr} \bmod p$

=m*$g^0 \bmod p$

=m mod p

=m

The procedure gives same result as that of traditional Elgamal cryptosystem and adds additional strength to it through factorization hardness problem. Factorization hardness – Given g^{xkr}, it is hard to factorize x, k and r.

b) Image Decryption

The decryption procedure for image is illustrated in Figure 4.

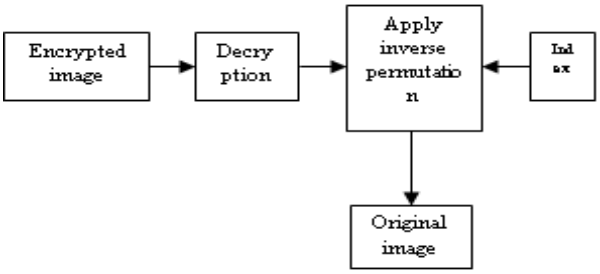


Fig 4 *Decryption Procedure*

The encrypted image is first decrypted using Blowfish decryption algorithm. Next, with the permutation order that is stored in index, inverse permutation is applied to the image blocks to obtain the original image.

C. Homomorphism

The major problem that leads to data breach is when the data is being outsourced for analysis. The process of outsourcing data for analysis demands revealing the original data to the third party. To address this issue, a property called homomorphism can be employed. The Homomorphic property enables the data owner to just reveal the cipher text rather than the plain text. There are algorithms that support homomorphism and Elgamal is one such. It is observed that the modified Elgamal stated in this contribution also supports homomorphism. Elgamal supports multiplicative homomorphism.

Proof: Let m_1 be the first plain text and m_2 be the second plain text.

$$E(m_1) = (g^{k_1 r_1}, m_1 y^{k_1 r_1})$$
$$E(m_2) = (g^{k_2 r_2}, m_2 y^{k_2 r_2})$$
$$E(m_1) * E(m_2) = (g^{k_1 r_1}, m_1 y^{k_1 r_1}) * (g^{k_2 r_2}, m_2 y^{k_2 r_2})$$
$$= (g^{k_1 r_1 + k_2 r_2}, m_1 m_2 y^{k_1 r_1 + k_2 r_2})$$
$$= E(m_1 * m_2)$$

III EXPERIMENT AND RESULT

The implementation is done in 2.71 GHz Intel core processor, 8.00 GB RAM and 64- bit OS. The result of the proposed scheme is compared with traditional Elgamal and modified Elgamal proposed in [7] and these schemes are also executed in the same configuration system. Different key sizes and data sizes are taken for analysis and it was observed that the running time is dependent on these two

factors. The running times are stated in milliseconds (ms) and all sizes are represented in terms of bits. The comparative results are shown in Table 1. It is observed that the execution time of the proposed work is twice as much as the execution time of traditional Elgamal and comparatively lower than the modified Elgamal proposed in [7]. Also it is clearly seen in the results of modified Elgamal that the execution time of decryption is more than the encryption time. The graphical representation of the above results is shown in the following Figures. Figure 5.1 shows the comparison of execution time for encryption for different key size and data size, Figure 5.2 shows comparison of decryption time. Total time is the sum of encryption time and decryption time.

IV CONCLUSION

Many reviews have been done on cloud security issues and some of the researches have proposed different ideas to address the issues in data security that prevails in cloud environment. Based on the fact that these solutions are not sufficient to face the cryptanalysis attacks, a novel approach to store patient data which is considered to be sensitive, has been proposed in this work. A modified Elgamal that supports homomorphism is also described. Though the computational complexity is little high than the existing algorithms, this scheme promises an added layer of security. A permutation preceded Blowfish technique has also been stated for image security. In future this can be extended to videos.

TABLE 1 *Comparison of Results*

Mod size(p) (In Bits)	*Key size (In Bits)*	*Data size (In Bits)*	*Traditional Elgamal*		*Modified Elgamal*		*Proposed work*	
			Encryption Time (In ms)	*Decryption Time (In ms)*	*Encryption Time (In ms)*	*Decryption Time (In ms)*	*Encryption Time (In ms)*	*Decryption Time (In ms)*
1024	*32*	*128*	*37*	*35*	*71*	*237*	*52*	*49*
1024	*32*	*256*	*41*	*36*	*89*	*210*	*78*	*51*
1024	*32*	*512*	*49*	*48*	*127*	*204*	*93*	*54*
1024	*32*	*1024*	*63*	*61*	*179*	*197*	*130*	*59*
1024	*64*	*128*	*43*	*39*	*95*	*258*	*87*	*83*
1024	*64*	*256*	*48*	*42*	*113*	*247*	*99*	*91*
1024	*64*	*512*	*54*	*51*	*133*	*239*	*115*	*97*
1024	*64*	*1024*	*71*	*69*	*194*	*214*	*147*	*117*
1024	*128*	*128*	*49*	*47*	*124*	*268*	*111*	*99*
1024	*128*	*256*	*57*	*55*	*143*	*245*	*125*	*101*
1024	*128*	*512*	*68*	*64*	*171*	*228*	*139*	*119*
1024	*128*	*1024*	*84*	*81*	*194*	*217*	*176*	*134*
1024	*256*	*128*	*53*	*49*	*137*	*263*	*132*	*112*
1024	*256*	*256*	*67*	*63*	*159*	*259*	*147*	*139*
1024	*256*	*512*	*77*	*71*	*168*	*240*	*121*	*142*
1024	*256*	*1024*	*93*	*89*	*203*	*236*	*136*	*164*
1024	*512*	*128*	*66*	*62*	*141*	*274*	*138*	
1024	*512*	*256*	*75*	*73*	*164*	*269*	*152*	
1024	*512*	*512*	*87*	*84*	*175*	*256*	*169*	*149*
1024	*512*	*1024*	*95*	*88*	*235*	*249*	*194*	*174*

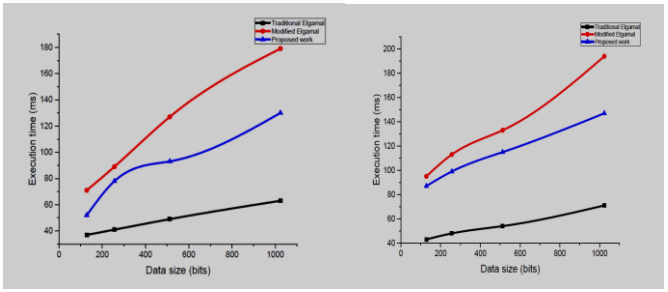


Fig 5.1 *(a): Key size: 32-bits* ***Fig 5.1****(b): Key size: 64-bits*

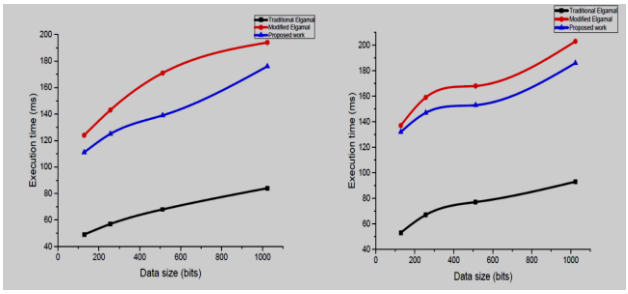


Fig 5.1 *(c): Key size: 128-bits* ***Fig 5.1*** *(d): Key size: 256-bits*

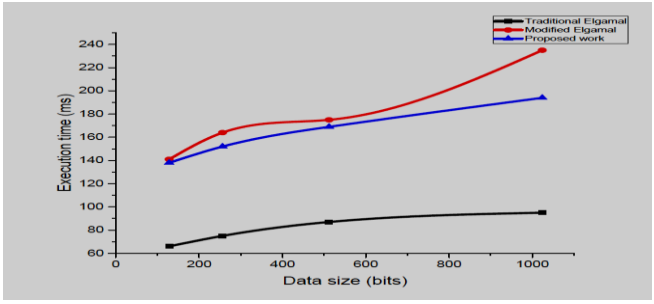


Fig 5.1 *(e): Key size: 512-bits*

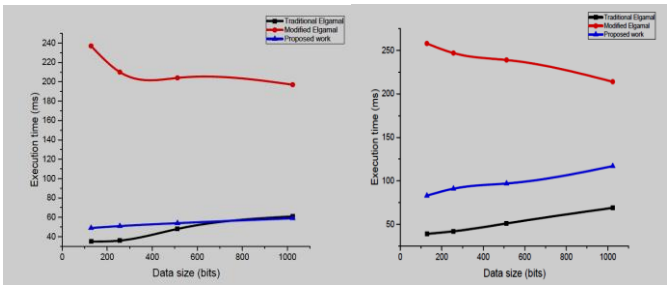


Fig 5.2 *(a): Key size: 32-bits* ***Fig 5.2*** *(b): Key size: 64-bits*

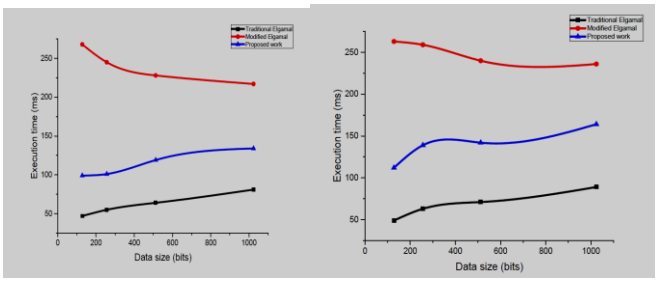


Fig 5.2 *(c): Key size: 128-bits* ***Fig 5.2*** *(d): Key size: 256-bits*

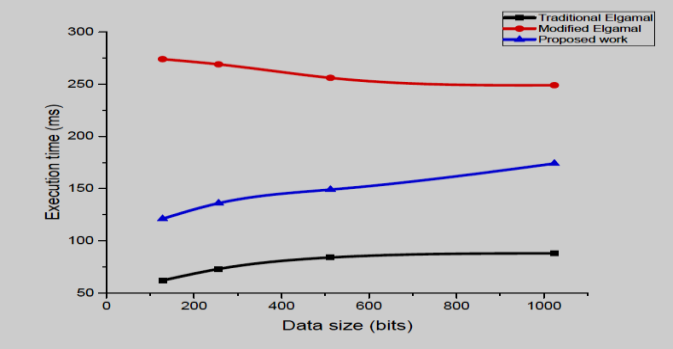
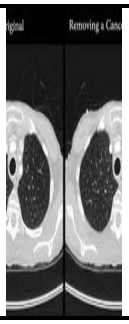
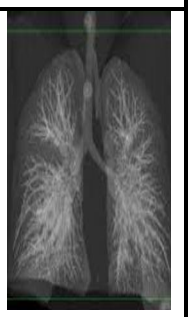
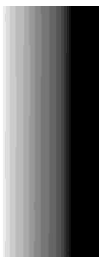
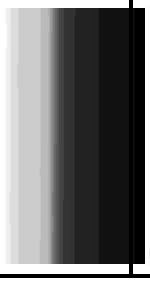
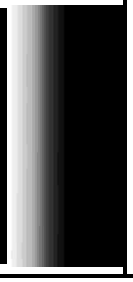
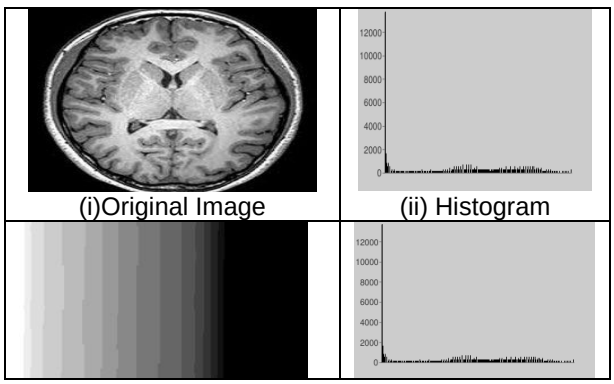


Fig 5.2 *(e): Key size: 512-bits*

TABLE 2 *Original images and their corresponding permuted images*

Image	Brain	Tumour cell	X Ray	Lungs
Original image				
Permuted image				

The histograms of original image and permuted image are shown below in Fig 6. It could be observed that the permutation does not change the image details.



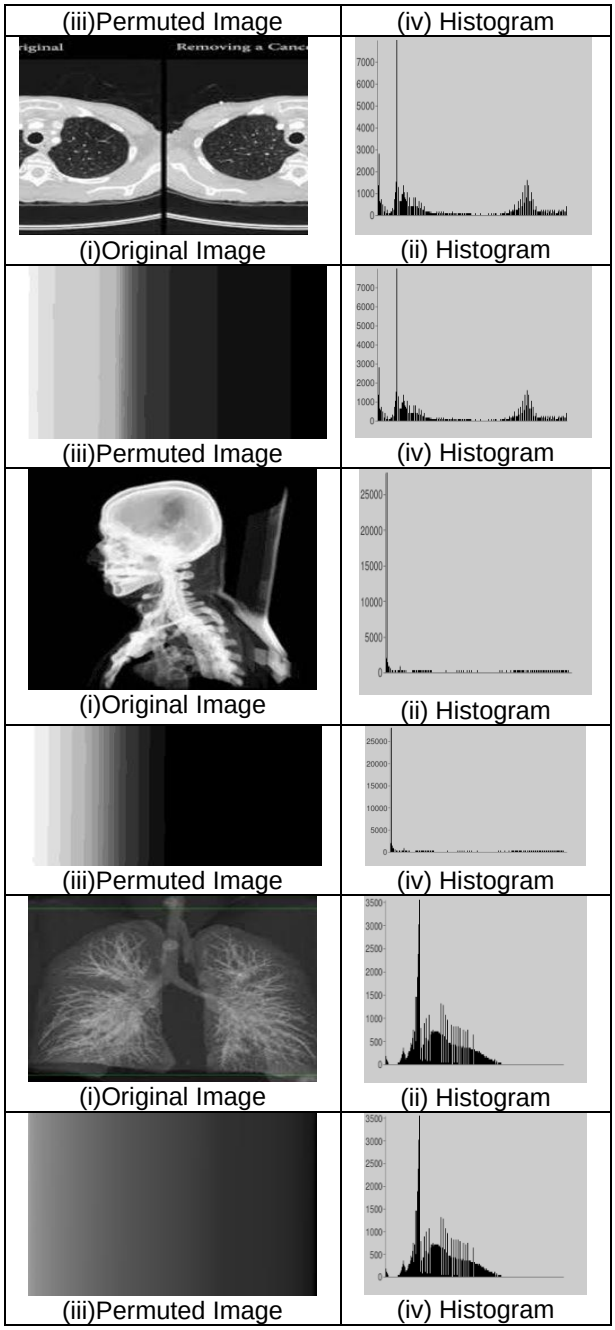


Fig 6 *Histograms of original image and permuted image*

REFERENCES

[1] Farhan Bashir Shaikh, Sajjad Haider, “Security threats in Cloud Computing”, 2011 International conference for Internet technology and secured transactions, 214 – 219, 2011.

[2] Sandeep K. Sood, “A combined approach to ensure data security in cloud computing”, Journal of Network and Computer Applications, Volume 35, Issue 6, November 2012, pp 1831 – 1838.

[3] Juels A, Burton J, Kaliski S. PORs: proofs of retrievability for large files. Proceedings of CCS ’07, p. 584–597, 2007.

[4] Pedro Ramos Brandao, “The Importance of Authentication and Encryption in cloud computing framework security”, International Journal on Data Science and Technology, Volume 4, Issue 1, 2018, pp 1-5.
[5] Mohit Kumar, Akshat Aggarwal, Ankit Garg, “A review on various digital image encryption techniques and security criteria”, International Journal of Computer Applications, Volume 96, Issue 13, 2014, pp 19 – 26.
[6] Geetanjali Sinha, Prabhu Shankar and Shaurya Jain, “Evolution of access control models for protection of patient details, a survey”, International Journal of Engineering and Technology, Volume 7, 2018, pp 554 – 558.
[7] Prashant Sharma, Sonal Sharma, Ravi Shankar Dhakar, “Modified Elgamal cryptosystem”, 2nd International conference on Computer and Communication Technology, 2011, pp 439 – 443.
[8] Pia Singh, Karamjeet Singh, “Image encryption and decryption using Blowfish algorithm in MATLAB”, International Journal of Scientific and Engineering Research, Volume 4, Issue 7, 2013, pp 150 – 154.
[9] D. Descher, M., Masser, P., Feilhauer, T., Tjoa, A.M.andHuemer, “Retaining data control to the client in infrastructure clouds,” Int. Conf. Availability, Reliab.Secur. (pp. 9-16). IEEE., 2009, pp. 9–16.
[10] E. Mohamed, “Enhanced data security model forcloud computing,” Informatics Syst. (INFOS), 8th Int. Conf., 2012, pp. 12–17.