ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 5, May 2018

Implementation of Cloud based Image Scaling and Cropping using Modified Paillier Cryptosystem

Neha Tayade[1], P.M. Chouragade[2]

P.G Student, Department of Computer Engineering, Government College of Engineering, Amravati, India[1]

Assistant Professor, Department of Computer Engineering, Government College of Engineering, Amravati, India[2]

ABSTRACT: The progression of cloud computing and increase in image size are making an attractive business model for the outsourcing of image storage and image processing. There are many advantages of outsourcing, like it ensures data confidentiality in the cloud. In this paper, we implement a modified paillier cryptosystem algorithm along with scaling and cropping operation. Instead of multiple users can view a complete image, user can just see the portion of that image which is cropped and also user can zoom it by scaling operation for better view. This paper reviews the analysis and results show that by using scaling and cropping operation it is space efficient as well as it is minimize the overhead. Its shows that our proposed system is efficient, effective and secure .

KEYWORDS: Image Processing, Secret Image Sharing, Cloud Computing, Image Scaling and Image Cropping.

I. INTRODUCTION

Cloud computing, the new term for the long dreamed vision of computing as a utility, enables convenient, on-demand network access to a centralized pool of configurable computing resource that can be rapidly deployed with great efficiency and minimal management overhead. The amazing advantages of Cloud Computing include: On-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing, and transference of risk. Thus, Cloud Computing could easily benefit its users in avoiding large capital outlays in the deployment and management of both software and hardware. As Cloud computing becomes increasingly popular; more people are inclined to outsource their data to the cloud, due to its exibility and unlimited resources. In addition, it can reduce local data maintenance costs and offer a convenient communication channel to share resources among the data owner and legitimate data users.

Digital imaging is being increasingly used in a variety area of daily life such as medicine, personal photography, teaching and learning etc. Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or features associated with that image.Now days, image processing is rapidly growing technology, for that purpose it has become very important to secure them from leakages. The requirement of fulfill the security needs of images have led to development of the good encryption technique.

II. PROPOSED METHODOLOGY

The amount of digital images has exploded in recent years due to the proliferation of digital imaging devices with increasing resolution. Individuals and organizations are beginning to rely on third party datacenters to store, process, share, and manage images. The amount of digital images has exploded in recent years due to the proliferation of digital imaging devices with increasing resolution. Individuals and organizations are beginning to rely on third party datacenters to store, process, share, and manage images.

III. Proposed algorithm

A. *First Level Encryption Algorithm:*

```
Read input image
Set BufferedImageimg =Input image
Set BufferedImageimgout=null
Set newpos[]={0,1,2,......,img.width-1}
Shuffle newpos[]
Set j1=0;
For i= img.width to 0
    For j= img.height to 0
        Imgout[newpos[cnt]][j1]=img[i][j]
    End for
End for
Store encrypted image on server temporarily.
Store newpos[] as secrete key in database
```

B. *Tile level Encryption Algorithm*:

```
Read Encrypted image bytes
Set byte[] b=read encrypted image bytes
Set len=b.len
Set klen=b.len/2
Set enclen=b.len-(b.len/2)
if(b.length%2==0) then
    Set l=b.length;
Else
    Set l=b.length-1
End if
```

```
•   Set cnt=0
•   For i=1 to l
        •   byte b1=b[i]
        •   byte b2=b[i+1]
        •   int tem1=(int)b1
        •   int tem2=(int)b2
        •   int res=tem1^tem2
        •   byte bbb=(byte)(0xff & res)
        •   bkey[cnt]=bbb
        •   benc[cnt]=b[i]
•   End For
•   if(b.length%2!=0)      then
        •   benc[cnt]=b[b.length-1]
        •   Write benc in file and store on server
        •   Write bkey[] in file and store on server
```

C. *Tile level Decryption Algorithm :*

```
•   Read Tile Encrypted image bytes
•   Set byte[] b=read encrypted image bytes
•   Set byte[] bkey=read key from server
•   Set byte[] out=new byte[orlen]
•   if(b.length%2==0) then
        •   Set l=b.length;
•   Else
        •   Set l=b.length-1
•   End if
•   Set cnt=0
•   For i=1 to l
```

```
        • byte b1=b[i]
        • byte b2=bkey[i]
        • int tem1=(int)b1
        • int tem2=(int)b2
        • int res=tem1^tem2
        • byte bbb=(byte)(0xff & res)
        • out[cnt]=b[i]
        • out[cnt+1]=bbb
        • i=i+1
    • End For
    • if(b.length%2!=0)    then
        • out[cnt]=b[b.length-1]
        • Write out in file and store on server
```

D. *First Layer Decryption Algorithm:*

```
    • Read partial decrypted image after tile level decryption
    • Set BufferedImageimg =Input image
    • Set BufferedImageimgout=null
    • Set newpos[]=Read Key From DB()
    • Set j1=0;
    • For i= img.width to 0
        • For j= img.height to 0
            • imgout[i][j]= img[newpos[cnt]][j1]
        • End for
    • End for
    • Store decrypted  image on server temporarily
```

E. *Image Scaling :*

```
    • View decrypted image
    • Set BufferedImageimg=partial decrypted image
```

- Input scaling factor eg 2,3,4,5…
- Set n=scaling factor
- Find new height and width
- Height=img.height*n
- Width=img.width*n
- Scale img with new height and width
- Decrypt scaled image

F. *Image Cropping:*

- View decrypted image.
- Set BufferedImageimg=partial encrypted image.
- Input 4 x,y coordinates of image.
- Width=x2-x1;
- Height=y4-y2
- BufferedImage b=Get subpart of img .
- Decrypt b and store on server temporarily.

IV. EXPERIMENTAL EVALUATION

In this section, we present the experimental evaluation of the proposed system. We investigated the success of our scheme in the context of encryption and decryption. To perform our evaluation, we use result of existing and proposed scheme. In existing system user apply paillier cryptosystem algorithm in color or black and white images.They perform scaling and cropping operation on encrypted images. With paillier tile based system, we require more encryption/decryption operations to be performed. But in case of our proposed system we require lessencryption/decryption operations. Below figuresresult will show the size of images after encryption and decryption operation. To make it practical, we propose some improvements to decrease overheads resulted from existing system.

In 2DCrypt, we put a number of bytes in a tile, and encrypt the tile instead of encrypting each pixel independently. Furthermore, we optimized the Paillier scheme to limit its storage requirement. Due to these improvements, 2DCrypt required less cloud storage than the naive per-pixel encryption. The computational overhead is also significantly reduced because of fewer encryptions and decryptions rounds. However, are dependent on the image size and the user's scaling and cropping parameters.Graphical representation is given below.

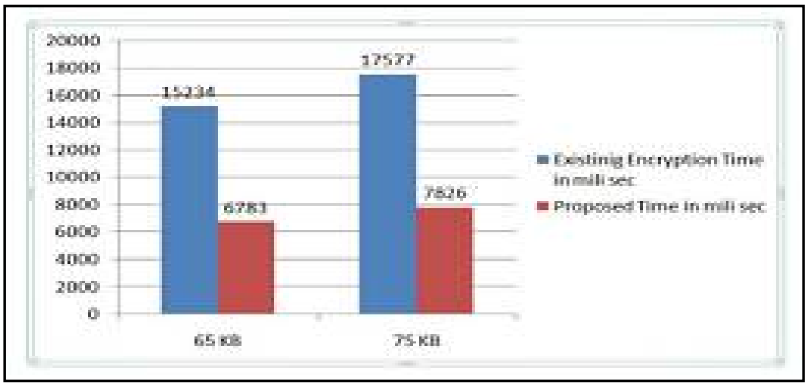

Figure 1: Time Parameter

The above figure shows the graphical representation of proposed system. In which graph shows the time parameter for encryption and decryption .Where blue lines shows existing encryption time mili second, and red lines shows proposed system encryption time in mili second. In which figure shows that the proposed system is efficient than the existing system.

V. CONCLUSIONS

Cloud-based image processing has data confidentiality issues, which can lead to privacy loss. In this, we addressed this issue by proposing Modified Pailler Cryptosystem in which we use Histopathology image scheme that allows a cloud server to perform scaling and cropping operations without learning the image content. This paper shows implementation and evaluation of cloud based image scaling and cropping using modified paillier cryptosystem. The purposed of our propose system is to develop a secure cloud based image storage and editing application using Modified Paillier Cryptosystem.As compare to existing system in propose system it will increase performance of the existing system by minimizing the overhead.

REFERENCES

1. M. Mohanty , M. R. Asghar , and G. Russello , “2DCrypt: Image Scaling and Cropping in Encrypted Domain”, 2016.
2. C. Gentry, “A fully homomorphic encryption scheme,” Ph.D. dissertation, Stanford University, Stanford, USA, 2009.
3. M. Mohanty, W. T. Ooi, and P. K. Atrey, “Scale me, crop me, know me not: supporting scaling and cropping in secret image sharing”, in Proceedings of the 2013 IEEE International Conference on Multimedia & Expo, San Jose, USA, 2013.
4. K. Kansal, M. Mohanty, and P. K. Atrey, “Scaling and cropping of wavelet-based compressed images in hidden domain”, in MultiMediaModeling, ser. Lecture Notes in Computer Science, Volume No. 8935, pp. 430–441, 2015.
5. C.-C. Thien and J.-C. Lin, “Secret image sharing,” Computers and Graphics, vol. 26, pp. 765–770, October 2002.
6. C.-Y. Hsu, C.-S. Lu, and S.-C. Pei, “Image feature extraction in encrypted domain with privacy-preserving SIFT,” IEEE Transactions on Image Processing, vol. 21, no. 11, pp. 4593–4607, 2012.
7. D. X. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in IEEE Symposium on Security and Privacy, , pp. 44–55 ,2000.
8. P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in Advances in Cryptology EUROCRYPT, vol. 1592, pp. 223–238, 1999.
9. C.-C. Thien and J.-C. Lin, “Secret image sharing,” Computers and Graphics, vol. 26, pp. 765–770, October 2002.
10. G. Ateniese, K. Fu, M. Green, and S. Hohenberger, Improved proxy re-encryption schemes with applications to secure distributed storage,” ACM Transactions on Information and System Security, vol. 9, pp. 1–30, February 200