

Secured Document Sharing Using Visual Cryptography in Cloud Data Storage

K. Brindha, N. Jeyanthi

*School of Information Technology and Engineering, VIT University, Vellore 632014, Tamilnadu, India
Emails: brindha.k@vit.ac.in njeyanthi@vit.ac.in*

Abstract: *Security has emerged as the most concerned aspect of cloud computing environment and a prime challenge for the cloud users. The stored data can be retrieved by the user whenever and wherever required. But there is no guarantee that the data stored in the cloud server has not been accessed by any unauthorized user. The current cloud framework does not allow encrypted data to be stored due to the space and storage cost. Storing secret data in an unencrypted form is vulnerable to external attacks by both illegitimate customers and a Cloud Service Provider (CSP). Traditional encryption techniques require more computation and storage space. Hence, protecting cloud data with minimal computations is the prime task. Secured Document Sharing Using Visual Cryptography (SDSUVC) technique proposes an efficient storage scheme in a cloud for storing and retrieving a document file without any mathematical computations and also ensures data confidentiality and integrity.*

Keywords: *Security, computing, environment, hindrance, frame work, vulnerable, attackers, efficient, visual cryptography, technique, confidentiality, integrity.*

1. Introduction

The ever increasing demand for large scale computing, combined with advances in minimum cost and fast networking technologies, has helped cloud computing to emerge as a promising and assuring computing model. In contrast to traditional IT services, it distinguishes itself as an exclusive high-performance Internet based technology which is economically feasible, both in terms of setup and maintenance. A cloud provides essential services, such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) to the customers. Even though the benefits of using the cloud are clear and understood, some of the problems remain unsolved. With the files stored on the outsourced storage, organizations and individuals are no longer worried about the size of the computer's

hard disk, as well as the risk of losing their important documents due to system crash. The various cloud storage service providers, such as Amazon, Microsoft, Google, Apple, etc. provide different storage services to users for keeping their important documents and files securely on the remote storage. Storage service cost of storage and other additional features differ from one provider to another but they provide certain amount of free space to users. Generally users easily store their files in Google drive, Drop box, One drive, Sky drive, etc. with their mail id and password. Most of the storage providers store the user files in unencrypted form.

The major hurdle to existing cloud services is security. The issue of security in cloud computing has serious attention in academics [1, 2], industry [3] and government [4]. The user can overcome this problem by encrypting the data before storing it in a cloud storage and retrieving it by decryption. However, especially business users use traditional symmetric algorithms, such as DES, AES, Blowfish, etc. to encrypt their sensitive document before storing it on the offsite storage. But the time, storage and mathematical computation are more in traditional technique. MS word is used for the prime task of creating documents, such as letters, brochures, tests, project documentations, assignments etc. due to its simplicity, easy accessibility and adaptability. Hence, potential users store their information in the word document.

This paper proposes a novel method called Secured Document Sharing Using Visual Cryptography (SDSUVC) technique for efficient document storage, which requires less storage space on cloud and needs less time complexity for the retrieval of original document using this technique.













The remainder of this paper is organized as follows. Section 2 covers a review of related work, Section 3 describes the architecture of SDSUVC technique, Section 4 illustrates encryption, Section 5 discusses the decryption process, Section 6 analyses the experimental result, Section 7 covers complexity and Section 8 concludes the paper.

2. Related work

Visual Cryptography was developed by Moni Naor and Adi Shamir in 1994 at the Eurocrypt Conference. Visual cryptography is “a new type of cryptographic technique, which can decode concealed images without any mathematical computations.” This allows anyone to use the system without any knowledge of cryptography scheme and without need of any computations whatsoever. The basic model of visual cryptography uses a binary image which consists of black and white pixels and each pixel is handled separately. Each original pixel exists in n shares and each share is a collection of m black and white sub pixels. The resultant image can be described as n out of m Boolean matrices $S = [S_{ij}]$ where

- $S_{i,j} = 1$ if the j -th subpixel in the i -th share is black
- $S_{i,j} = 0$ if the j -th subpixel in the i -th share is white
- Combine the shares s_1, s_2, \dots, s_n , which properly align the sub pixels to get the original image.

Table 1. Encoding and stacking of a pixel

PIXEL	WHITE		BLACK	
Probability	50%	50%	50%	50%
Share – I				
Share – II				
Stack Share I & II				

To illustrate the concept of Visual Cryptography, the simplest version of the “two out of two” scheme, where each original pixel of the secret image is coded into a pair of subpixels in each of the 2 shares. It is very difficult to identify the information stored in the shares. The drawback is, that the retrieved secret images are two times larger than the original image and the quality of image [5].

The basic two out of two visual cryptography techniques can be extended to k out of n schemes [5-8]. A more general model can recover the image and the participant from the forbidden set cannot access the useful information. This model reduces the pixel expansion but it produces only optimal contrast of the image [9]. The grey level image is transformed into approximate binary image by dithering technique. The existing scheme applied to binary image is used to create the image shares [10].

For more security purpose, Visual Cryptographic technique is applied to colour images [11].

Conventional Visual Cryptography technique generates a noisy pixel on shares which indicates that some hidden secret images are on a share. This can be avoided by Extended Visual Cryptography technique [12].

All the previous schemes have dealt with sharing of merely one secret. The extension of the scheme is trying to hide the multiple secrets instead of a single secret. The merit of this scheme is its ability to hide more than one secret within a set of secrets. The limitation of this scheme is the resultant relatively optimum size and poor quality of the image [13, 14]. All the existing cryptographic techniques have been so far applied only to an image and not for a document file.

3. SDSUVC architecture

SDSUVC technique mainly uses Visual Cryptography to protect a secret document file in the cloud. In all the current work in the domain of cloud computing, security is focused on using some conventional encryption algorithm for storing and retrieving data. The traditional encryption technique requires more time, space and also involves complex computations. Hence the proposed SDSUVC approach is to avoid the use of conventional encryption techniques, instead it uses Visual

Cryptography for uploading and downloading secret data. The overall concept of the system is very simple and it also protects the secret in the document. For a security purpose, instead of uploading the original document file, it must be converted into a text file, again into an image files and then uploaded in to the cloud.

Later the downloaded image files must be converted into a text file and again into the original document file. The following steps are to be performed for uploading a document file (doc/docx) into a cloud and downloading a document file from the cloud as shown in Fig. 1.

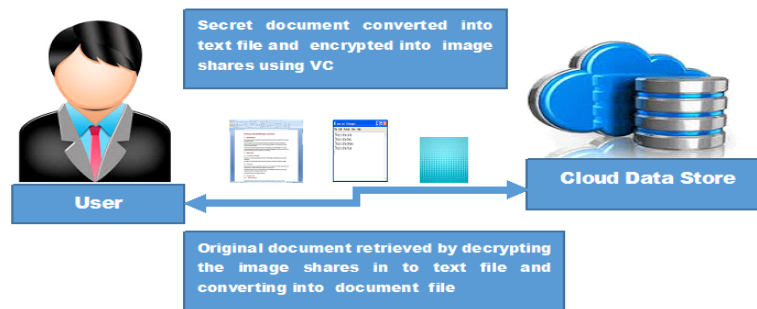


Fig. 1. SDSUVC architecture

Uploading a secured document file

- Select the document file which is to be uploaded
- Convert the document file (doc/docx) into a text file
- Convert the text file into image files using SDSUVC technique
- Upload the the image files on a cloud

Downloading a secured document file

- Select the image files which are to be downloaded
- Download the image files from the cloud
- The resultant text file can be obtained by stacking the image files using SDSUVC technique
- Convert the text file into the original document file

4. SDSUVC encryption process

When the user wants to upload a document file, which contains some secret information, it must be encrypted by SDSUVC encryption technique which involves two phases. In the initial phase the document file must be converted into a text file using Apache Poi application programming interface and in the next phase the resultant text file must be encrypted using SDSUVC encryption technique. Every line from the text file must be read and each and every character converted into an integer (ASCII value). An individual pixel must be fed on a buffered image using SetRGB method. A line of a pixel must be stored in the first image and the next one in the second image and the other line in the third image. This process is repeated until the file comes to an end. Finally the shares of the image file must be uploaded into a cloud as shown in Fig. 2.

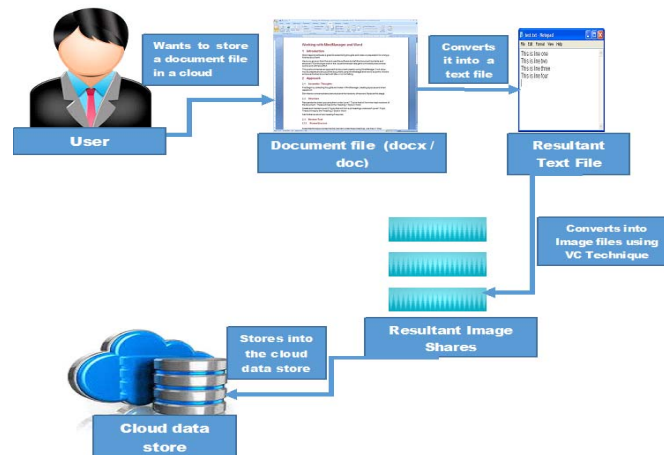


Fig. 2. Document file stored as image files in a cloud

Algorithm for SDSUVC encryption – conversion of an original document file into a text file

- Read the document file with doc/docx extension as “rama.docx”
- Use HWPf (Horrible Word Processor Format) document component to access the MS-Word document file
- The class org.apache.poi.xwpf.extractor.XPFWordExtractor extracts and returns simple data from a Word file
- Use getText() function to retrieve all the text from the document
- Store the data in the text file “thetextfile.txt”

Encryption of a text file into image shares

- Open the text file “Secret.txt” a in read mode
- Create three image objects Image1, Image2 and Image 3 belonging to the Buffered image class
 - Read a text from the character input stream
 - Initialize the dynamic string array “list”
 - Read each line from the text file and store it in the list
 - Compute the value for width and height of an image
 - Width = No of characters in a line
 - Height = No of lines in a file
 - Height and Width intialized to image1 , image2 and image3.
 - Read each line from the dynamic array list
 - Convert each character in a line into an integer (ascii value)
 - Setting an individual pixel on a buffered image using SetRGB method.
 - Store one line of pixels in image1 , the next line of pixels in image2 and another line in image3
- This process is repeated until the end of the file
- Finally save image1, image2 and image3 into a png format, using ImageIo write method

5. SDSUVC decryption process

When the user wants to retrieve the original document file from the image stored on the cloud, using SDSUVC decryption technique it involves two phases. In the initial phase the image file is converted into a text file. Every line from the image share which is in .png format must be read and every pixel from a line must be extracted using getRGB method. Each pixel must be converted into hexa code and the resultant character stored in a string buffer. This process is repeated until the file comes to an end. Finally all the contents of the buffer must be transferred into a text file and in the next phase the text file is converted into the original document file using Apache Poi application program interface as shown in Fig. 3.

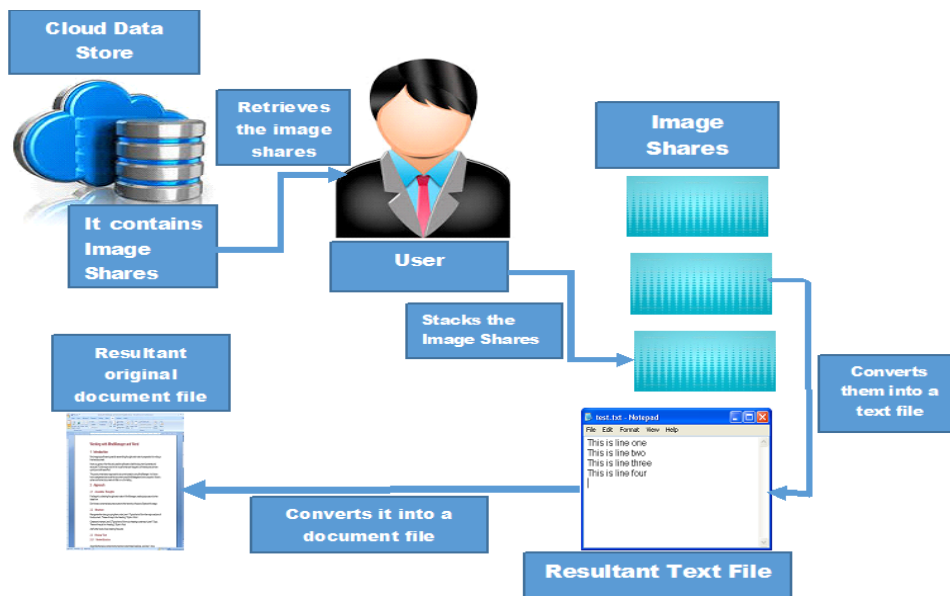


Fig. 3. Document file retrieved from the image files

Algorithm for retrieving the original document file from the Image shares Conversion of the image shares into a text file

- Read the Image file image1.png, image2.png image3.png using ImageIo.read method
- Create a new text file "Text.txt"
- Create a new buffer "bw" which is used to store the text to a character output stream
- Read each line from image1 , image2 and image3 file
- Extract each pixel value in a line using getRGB method
- Convert a pixel value to hexa code (ascii value)
- Store each character in a buffer "bw"
- Store the content of the buffer into a file "Text.txt" using the FileWriter method

Conversion of the text file into the original document file

- Use a file reader to read the source file
- Store all the content to a buffered reader
- Create a MS word document using XWPFDocument component
- Read each and every line from the buffer
- Create a document using XWPFDocument component
- Create a paragraph using XWPFParagraph component
- Create a run object in the paragraph using XWPFRun
- Set the font size and style with setFontFamily(), setFontSize() function
- Store the text in the document using setText() function
- This process is repeated until the file comes to an end
- Finally write the content of the document to the output file

6. Experimental analysis

The proposed SDSUVC algorithm has been implemented in Java and various experimental tests have been carried out. The algorithm is implemented in various sizes of document files; the performance of the algorithm is tested with parameters like size of the image shares and execution time for SDSUVC encryption and decryption technique with symmetric encryption algorithm, such as DES and AES. During the encryption process the original document file is converted into a text file and then converted into image shares.

The test result of the document file for the proposed SDSUVC technique is as follows:

Conversions of a document file into a text file

The document file is converted into a text file using apache poi application program interface. Fig. 4a and b show the sample document file and the resultant text file after the conversion.

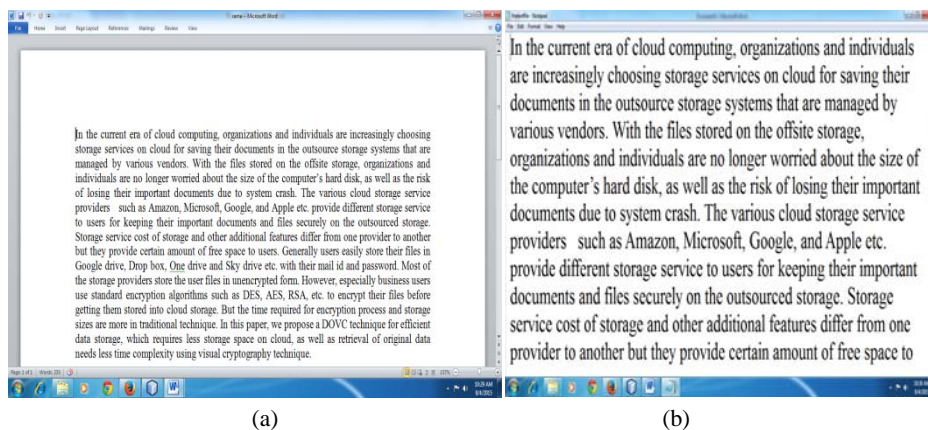


Fig. 4. Document file (.docx extension) (a); resultant text file (b)

Conversion of a text file into image files using SDSUVC technique

The resultant text file is encrypted into image files using SDSUVC technique. Fig. 5a, b, c shows the result of image files after the encryption process.

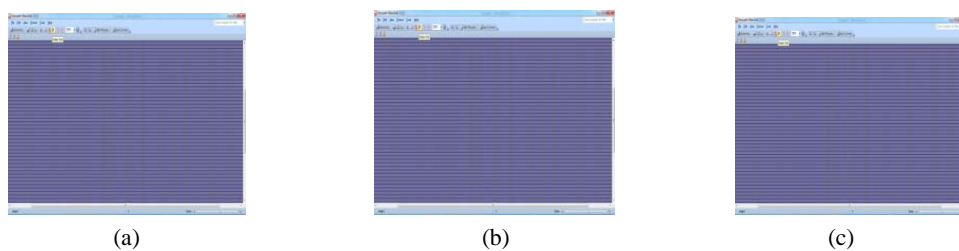


Fig. 5. Image file-1 (a); image file-2 (b); image file-3 (c)

Conversion of image files to a text file to original document using SDSUVC decryption technique

The resultant image files are decrypted into a text file using SDSUVC technique. The resultant text file is converted into a document file using apache poi application program interface. Fig. 6 shows the result of the original document file after conversion.

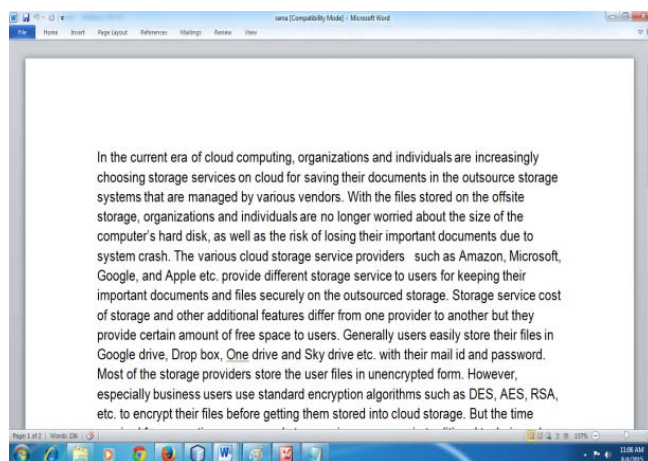


Fig. 6. Original document file

This technique has been tested with various document files and it is found that the size of the image share is quite less when compared with the original file during the encryption process. Table 2 depicts the various size text files compared with the image shares and Fig. 7 compares the size of the original document file with image shares.

Table 2. Result of the size of image shares during SDSUVC encryption

Size of an original document file (Kb)	Size of image shares (Kb)
14	3
15	3
17	3
21	6
25	6
29	6
33	9

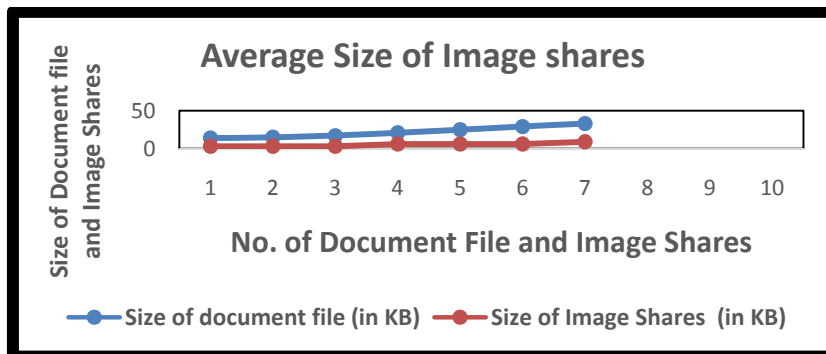


Fig. 7. Average size of image shares during encryption

The Size of the original document file is compared with the size of the decrypted document file and Fig. 8 shows the analysis of the size of the original document file with the decrypted document file. Based on the experimental result, it is found that both the document files are of the same size.

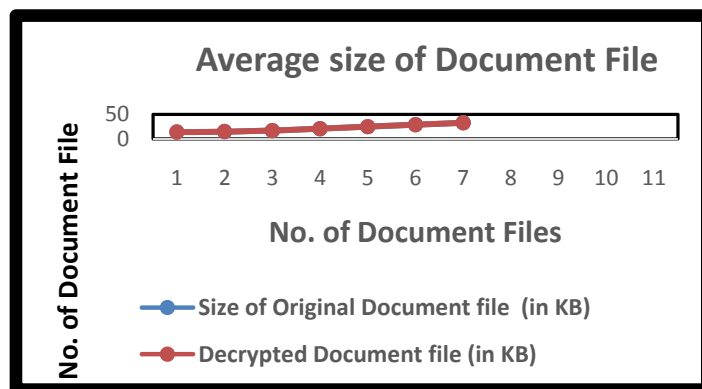


Fig. 8. Average sizes of a retrieved document file during decryption

The execution time is considered as the time taken to convert the document file into a text file and then into the image shares during encryption and the image shares into the text file and then into the original document file during decryption. These results are compared with traditional DES and AES algorithms. Table 3 compares the encryption time for SDSUVC technique with DES and AES. Fig. 9 shows the average execution time for these three algorithms.

Table 3. Execution time for Encryption in SDSUVC, DES and AES

Size of Original document file (Kb)	Execution time for SDSUVC (ms)	Execution time for DES (ms)	Execution time for AES (ms)
14	720	945	1000
15	807	962	1010
17	873	1043	1112
21	987	1167	1200
25	1192	1382	1500
29	1580	1580	1762
33	1414	1614	1800

The time required for encryption in SDSUVC technique is found to be less when compared with traditional DES and AES algorithms. The result is shown in Fig. 10.

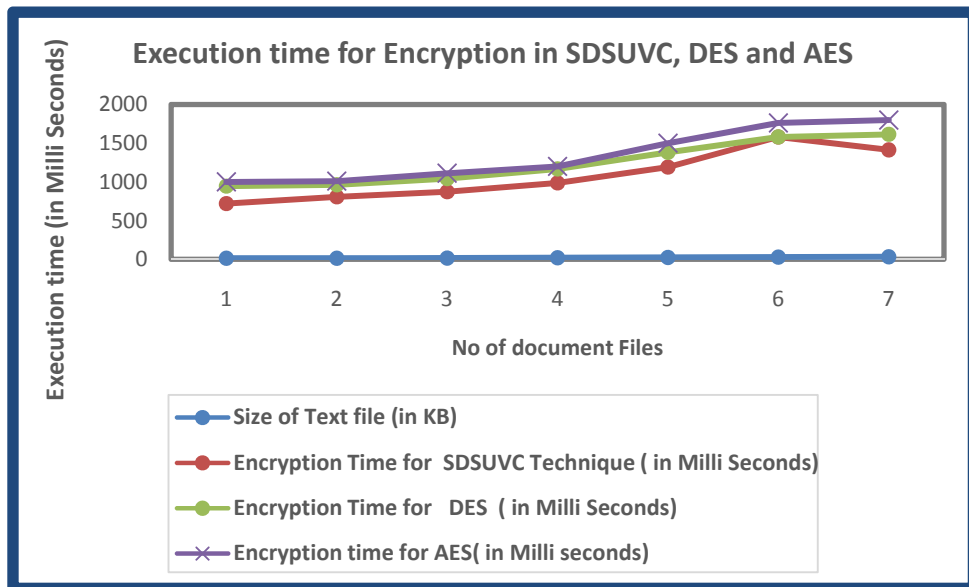


Fig. 9. Average execution time for encryption

The average execution time for the decryption process in SDSUVC technique is compared with the same in DES and AES algorithms. The decryption time of SDSUVC technique is found to be less when compared with DES and AES. The result is shown in Fig. 10.

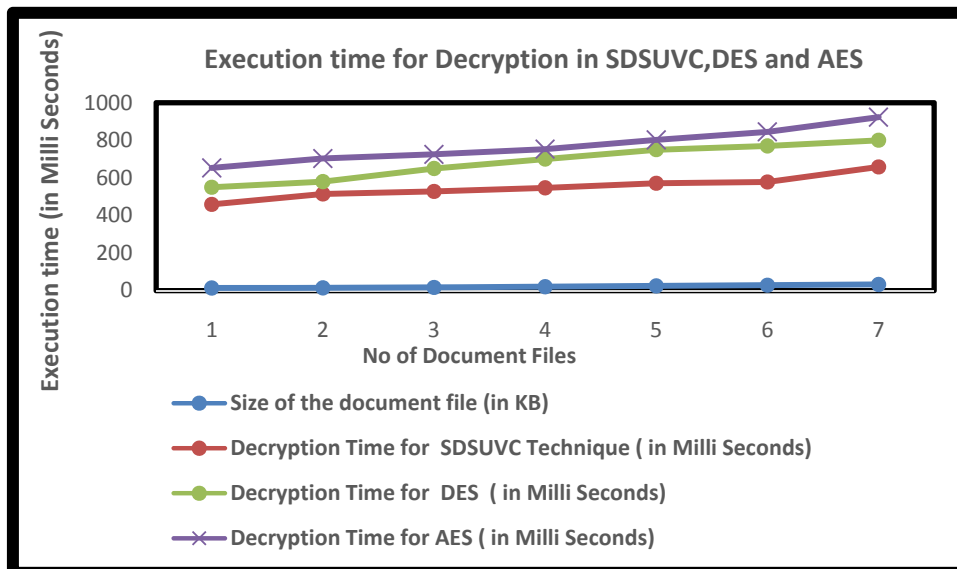


Fig. 10. Average execution time for decryption

Table 4 compares the size of the encrypted file in the SDSUVC technique with that of the files tested by standard DES and AES algorithms.

Table 4. Result of the size of encrypted files in SDSUVC, DES and AES

Size of original document file (Kb)	Size of image shares (SDSUVC encryption) (Kb)	Size of the encrypted file (DES) (Kb)	Size of the encrypted file (AES) (Kb)
14	3	14	14
15	3	15	15
17	3	17	17
21	3	21	21
25	6	25	25
29	6	29	29
33	6	33	33

The size of the encrypted file is very much reduced in SDSUVC algorithm when compared with the other two algorithms. The result is shown in Fig. 11.

7. Complexity of SDSUVC

Many of the existing schemes in Visual cryptography result in size of shares growing very large, depending on the image type and size. Typically, as the contrast improves, the share size also increases quite dramatically. This increases the image processing time to overall increase in the complexity of the schemes. It reduces the overall potential for the practical application of VC. The share sizes become completely unmanageable, specifically when high resolutions are used to share information. To put it in a nutshell, all these schemes state that hiding only a small amount of information within the shares has proven to be efficient. If a larger amount of data is required to be hidden, the share size becomes large and difficult to manage. Tracking this complexity has been a real challenge within VC.

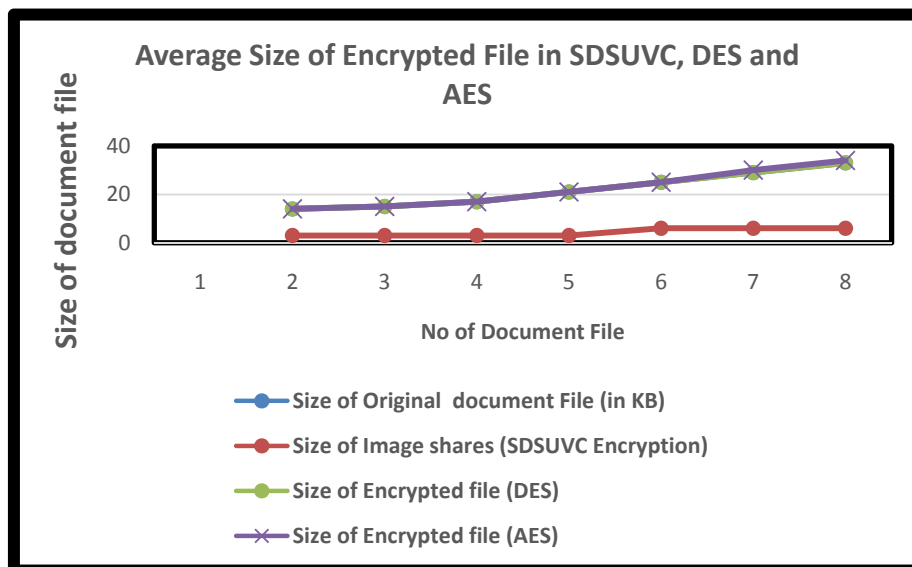


Fig. 11. Average size of an encrypted file

There are a number of schemes which present near optimal solutions for share sizes [15-17]. But the proposed SDSUVC technique can hide a large amount of secret details in a document file, as well as reduce the size of the encrypted image file remarkably. The time complexity of the existing requires cubic time $O(n^3)$ [18], but the proposed approach requires only quadratic time $O(n^2)$.

Time complexity of SDSUVC encryption process

Time required for conversion from a document file to text file = $O(n)$
 Time required for conversion from a text file to image shares = $O(2*n + 3*n^2 + 12)$
 Overall complexity of the encryption process = $O(n^2)$

Time complexity of SDSUVC decryption process

Time complexity of SDSUVC decryption process
 Time required for conversion from image shares into a text file = $O(2*n^2 + n + 5)$
 Time required for conversion from a text file to a document file = $O(5*n)$
 Overall complexity of the encryption process = $O(n^2)$

8. Conclusion

We would like to present a new technique called Secured Document Sharing Using Visual Cryptography (SDSUVC) to achieve data privacy in a cloud computing model. The Cloud Service Provider is considered unreliable and the data must be hidden not only from an external attack but also from the cloud provider. No one can extract the hidden secret information from the cloud. The usage of standard encryption techniques demands are associated with key management and it further involves increase in size, time and computational primitives. The complexity of our approach is shown to be reasonable and much less than standard algorithms. The SDSUVC technique ensures data confidentiality and security along with integrity and reputation.

References

1. Armbrust, M., A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia. Above the Clouds: A Berkeley View of Cloud Computing. Technical Report No UCB/EECS-2009-28, University of California at Berkeley, USA, 10 February, 2009.
2. Buyya, R., C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic. Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. – In: Future Generation Computer Systems. Amsterdam, The Netherlands, Elsevier Science, 2009.
3. Kaufman, L. M. Data Security in the World of Cloud Computing. – IEEE Security and Privacy, Vol. 7, 2009, No 4, pp. 61-64.
4. Mell, P., T. Grance. Effectively and Securely Using the Cloud Computing Paradigm. NIST, Information Technology Laboratory, 2009.

5. Naor, M., A. Shamir. Visual Cryptography. – In: Proc. of Advance in Cryptology-EUROCRYPT'94. – In: Lecture Notes in Computer Science. Vol. **950**. Springer-Verlag, 1995, pp. 1-12.
6. Blundo, C., A. De Santis, D. R. Stinson. On the Contrast in Visual Cryptography Schemes. – J. Cryptol.: J. Int. Assoc. Cryptol. Res., Vol. **12**, 1999, No 4, pp. 261-289.
7. Blundo, C., P. D'Arco, A. De Santis, D. R. Stinson. Contrast Optimal Threshold Visual Cryptography Schemes. – SIAM J. Discrete Math., Vol. **16**, 2003, No 2, pp. 224-261.
8. Hofmeister, T., M. Krause, H. U. Simon. Contrast-Optimal k out of n Secret Sharing Schemes in Visual Cryptography. – Theoretical Computer Science., Vol. **240**, Jun 2000, No 2, pp. 471-485.
9. Ateneise, G., C. Blundo, A. De Santis, D. R. Stinson. Visual Cryptography for General Access Structures. – Inf. Comput., Vol. **129**, September 1996, No 2, pp. 86-106.
10. Yang, C. N. New Visual Secret Sharing Schemes Using Probabilistic Method. – Pattern Recognition Letter, Vol. **25**, 2004, pp. 481-494.
11. Hou, Y.-C. Visual Cryptography for Color Images. – Pattern Recognition, Vol. **36**, 2003, pp. 1619-1629.
12. Lee, K.-H., P.-L. Chiu. An Extended Visual Cryptography Algorithm for General Access Structures. – IEEE Transactions on Information Forensics and Security, Vol. **7**, 2012, No 1, pp. 219-229.
13. Chen, S.-K. A Visual Cryptography Based System for Sharing Multiple Secret Images. – In: ISCGAV'07. – In: Proc. of 7th World Scientific and Engineering Academy and Society (WSEAS) International Conference on Signal Processing, Computational Geometry and Artificial Vision, Stevens Point, Wisconsin, USA, 2007, pp. 117-122.
14. Feng, J.-B., H.-C. Wu, C.-S. Tsai, Y.-F. Chang, Y.-P. Chu. Visual Secret Sharing for Multiple Secrets. – Pattern Recognition, Vol. **41**, 2008, Issue 12, pp. 3572-3581.
15. Yang, C.-N., T.-S. Chen. Size-Adjustable Visual Secret Sharing Schemes. – IEICE Transactions 88-A(9), 2005, pp. 2471-2474.
16. Yang, C.-N., T.-S. Chen. New Size-Reduced Visual Secret Sharing Schemes with Half Reduction of Shadow Size. – IEICE Transactions 89-A(2), 2006, pp. 620-625.
17. Yang, C.-N., T.-S. Chen. Extended Visual Secret Sharing Schemes: Improving the Shadow Image Quality. – IJPRAI, Vol. **21**, 2007, No 5, pp. 879-898.
18. Jaafar, A. M., A. Samsudin. A New Public-Key Encryption Scheme Based on Non-Expansion Visual Cryptography and Boolean Operation. – IJCSI International Journal of Computer Science Issues, Vol. **7**, July 2010, Issue 4, No 2, pp. 1-10.