Securing Portable Document Format File Using Extended Visual Cryptography to Protect Cloud Data Storage

K. Brindha and N. Jeyanthi
(Corresponding author: K. Brindha)

School of Information Technology and Engineering, VIT University
Vellore 632014, Tamilnadu, India
(Email: brindha.k@vit.ac.in)

(Received Apr. 1, 2016; revised and accepted July 19 & Aug. 29, 2016)

Abstract

With the vast development in cloud computing model, various organizations and individuals often deploy the cloud without reviewing the security policies and procedures which can cause great risk in their business. Securing data in cloud storage becomes a challenging task not only for the cloud user but also to the Cloud Service Provider (CSP). Storing secret data in unencrypted form is susceptible to easy access to both the unauthorized people and the CSP. Standard encryption algorithms require more computational primitives, storage space and cost. Therefore protecting cloud data with minimal computation and storage space is of paramount significance. The Securing Portable Document Format file Using Extended Visual Cryptography (SPDFUEVC) technique proposes efficient storage to achieve data confidentiality and integrity verification with minimal computation, time complexity and storage space.

Keywords: Confidentiality; Cryptography; Data Integrity; Visual Cryptography

1 Introduction

Leading to dramatic change in the computing services, cloud computing has become very popular in IT industries. It attracts the attention of industry and academia alike. The main aim of cloud computing is to provide flexible, feasible and secure services to users of network [10, 21, 27, 37]. In the present revolutionary scenario of IT explosion, cloud computing faces the ever growing demand for large scale computing with minimum cost and fast networking technologies. It has to prove its economic feasibility both in terms of setup and maintenance [24, 39]. A cloud provides fundamental 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 persist and remain unsolved [36, 45, 46].

Enterprises and individuals who have stored their files on the cloud storage are worried neither about the storage space in the hard disk of the computer nor the risk of the loss of their valuable files due to computer crash. Different CSPs like Google, Amazon, Apple, Microsoft, etc. offer various storage services to customers for storing their valuable files safely on the external storage terms and cost of storage services and other special benefits vary from one service provider to another. They offer certain amount of free storage space to customers. Usually cloud consumers can safely store their files in One drive, Google drive, Sky drive, Drop box, etc. with their personal email address and password. PDF (Portable Document Format) is a file format that has in itself all the elements of a printed document as an electronic image. It is especially useful for documents such as medical records, financial data, tender quotations etc. Hence most of the potential users store their information in the PDF file but in unencrypted form. Security as the major threat to the above cloud storage system [43].

The issue of security in cloud storage is of great concern in the academics [9], the industry[20] and the government [25]. This problem can be overcome by enciphering the data before storing it in cloud storage and retrieving it by deciphering. However, especially commercial users use conventional encryption algorithm [3, 13, 23, 32, 41, 46] such as AES, DES, Blowfish, etc. to encrypt their confidential data before storing it in cloud storage. But the time complexity, storage space and cryptographic computation are enormous in these conventional algorithms. The specific security requirements in cloud storage are largely cloudy to the end users. The two main security threats are sensitivity and integrity of the data received from remote storage. In an earlier article we proposed Secured Document Sharing using Visual Cryptography (SD-

SUVC) technique for efficient document storage, which utilizes only less storage space and time complexity for the document retrieval and it provides data confidentiality [8] to some extent.

This paper proposes a novel method named Securing Portable Document Format file Using Extended Visual Cryptography (SPDFUEVC) for more efficient file storage with absolute data confidentiality and integrity. It requires only less storage space on cloud and less time complexity for the retrieval of original PDF file using this algorithm.

The remainder of this article is arranged as follows. Section 2 covers a study of related work, Section 3 describes the architecture of SPDFUEVC technique, Section 4 discusses encryption, Section 5 illustrates decryption, Section 6 analyses the experimental results, Section 7 covers computational complexity and Section 8 concludes the article.

2 Related Work

Visual Cryptography was invented by Naor and Shamir in 1994 at the Eurocrypt Conference. This new cryptographic method is perfectly secure which can encode and decode the secret image without any cryptographic computations [11, 23, 35]. This system divides the secret image into two shares such as cipher and key transparencies, which are indistinguishable from random noise. The original secret image is obtained by placing key transparency over the cipher transparency.

The basic model of visual cryptography uses binary image which consists of collection of black and white pixels handling each of them separately. The secret image is divided into 'n' shares and each pixel appears in 'n' shares. The resultant image can be described as 'n' out of 'm' Boolean matrix

$$S = s_{ij}$$

where, $S_{ij} = 1$, the j^{th} subpixel in the i^{th} share being black; $S_{ij} = 0$, the j^{th} subpixel in the i^{th} share being white. Combine the shares $s_1, s_2, ..., s_n$ which properly aligns the sub pixels to get the original image.

To illustrate the concept of Visual Cryptography, the simplest version of 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 specified in Figure 1. The drawbacks of this basic model are the huge size (the retrieved secret images two times larger than the original image) and the poor quality of the image [29].

The basic two out of two' visual cryptography techniques can be extended to $k \times n$ schemes [5, 7, 16, 17, 29, 44]. A more general model for visual cryptography based on general access structure and authorized and forbidden subsets of the participants has been developed. This model reduces the pixel expansion but it produces only optimal contrast of the image [4]. Hence the basic scheme in visual cryptography is restricted to binary image pattern which is insufficient in real time applications.

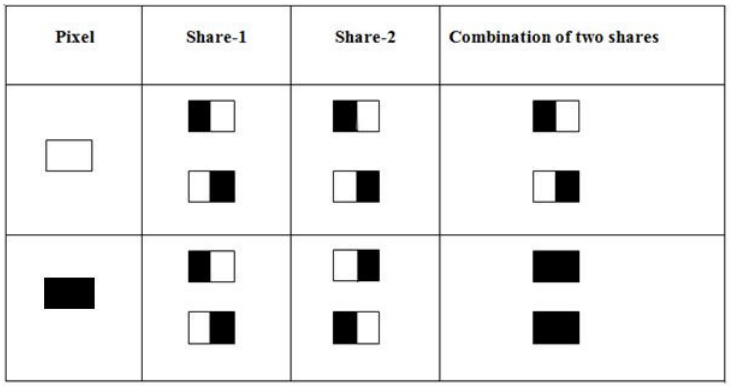

Figure 1: Encoding black and white pixels

The pixel in grey level images ranges from 0 to 255. The limitation of this method is pixel expansion and low contrast of resultant image [6]. Instead of using grey pixels directly into the image shares, the grey level image is converted into binary image by a dithering technique. The resultant binary image is applied to traditional visual cryptography scheme. This scheme reduces the pixel expansion but produces only optimum quality of the image [22]. The grey scale image is further enhanced by using half-toning method which converts it into binary image and then visual cryptography scheme is applied to resultant image. But this scheme is not suitable for a large sized secret image although there is an enhancement in contrast when compared with the previous case [26]. There is an enhancement only in visual cryptography being directly applied to grey scale image but the limitation of this scheme is low contrast of the image [26].

Most of the real time information contains color images and Visual Cryptographic algorithm is applied on them for securing the original information more effectively [18, 33]. Though this method reduces the pixel expansion it obtains only optimum quality of the image [2, 15].

Standard Visual Cryptography algorithm creates noisy pixel on image shares which shows that some secret information is embedded in them. This issue can be overcome by applying the Extended Visual Cryptography algorithm. The secret information hidden in these cover images cannot be easily identified by anyone other than the owner of the file [42]. This scheme is further enhanced by meaningful shares being generated by dithering technique [34].

All these previous schemes have dealt with sharing of merely one secret. Despite the merit of this scheme is its ability to hide more than one secret within a set of secrets, it has the limitation of large size and poor quality of the image [12, 14]. This scheme is further enhanced by sharing multiple secrets without pixel expansion and good quality of the image [17, 28, 30]. So far all the current visual cryptographic algorithms have been applied exclusively only on images and not on pdf files.

3 SPDFUEVC Architecture

SPDFUEVC (Securing Portable Document Format file Using Visual Cryptography) technique mainly uses Extended Visual Cryptography to protect a secret pdf file in cloud storage. In all the current work in the domain of cloud computing, security is focused on using conventional encryption algorithm AES (Advanced Encryption Standard) for storing and retrieving data [34, 35]. This traditional encryption technique requires more time, space and also involves complex computations. Therefore the proposed SPDFUEVC technique can effectively replace the use of conventional encryption algorithm by using the Extended 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 pdf file. For the security purpose, instead of uploading the original pdf file, it must be converted into a text file, next into image shares and then the resultant image shares, transformed into scrambled images and finally uploaded in to the cloud. Later the random noisy image shares should be extracted from the downloaded scrambled images and converted into the text file and again into the original pdf file. The following process is to be followed for uploading a pdf file into a cloud and downloading a pdf file from the cloud as shown in Figure 2.

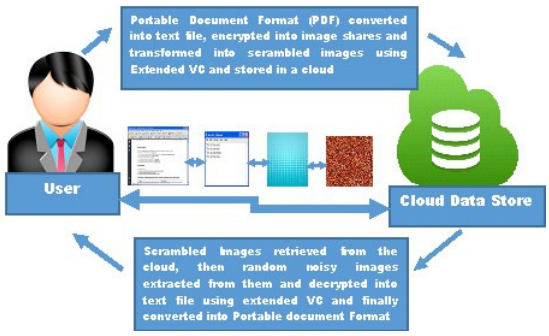

Figure 2: SPDFUEVC architecture

Algorithm 1 Uploading a secret pdf file

1: Select the pdf file which is to be uploaded
2: Convert the pdf file into text file
3: Convert the text file into image files and convert the resultant image files into scrambled image files using SPDFUEVC technique
4: Upload the scrambled image files on a cloud

4 SPDFUEVC Encryption Process

When a pdf file which contains some valuable data is to be uploaded, it has to be encrypted with SPDFUEVC encryption algorithm which involves three phases. In the initial phase, the pdf file must be converted into a text file using Apache PDFbox application programming interface and in the second phase the resultant text file must be encrypted using SPDFUEVC encryption technique. Each character in every line in the text file must be taken up and converted into an integer (ascii value). Then it is transformed into a pixel using SetRGB method and fed on a buffered image. Every pixel in a line must be stored alternately one in the first image and the next one in another image. This should be continued till the end of the file. Finally the image shares are converted into scrambled images and the scrambled images must be uploaded into a cloud as shown in Figure 3.

Algorithm 2 Downloading the Secret pdf file

1: Select the scrambled image files which are to be downloaded
2: Download the scrambled image files from the cloud
3: Extract image files from the scrambled image files
4: Obtain the resultant text file by stacking the image files using SPDFUEVC technique
5: Convert the text file into the original pdf file

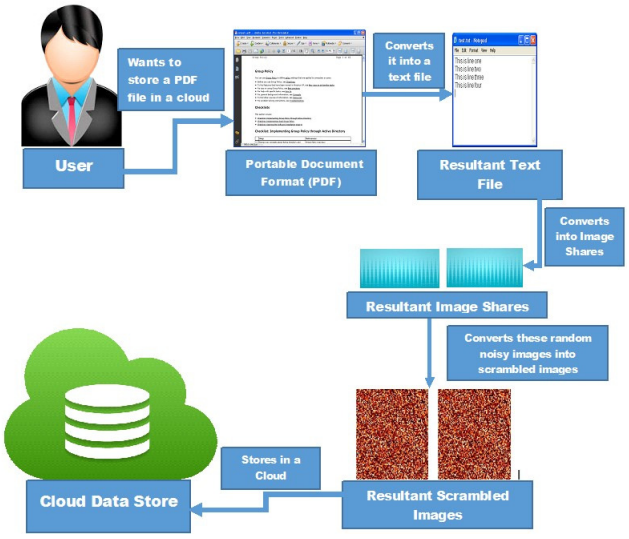

Figure 3: Secret pdf file in cloud storage

Algorithm 3 Conversion of original pdf file into text file

1: **Input: pdf file**
2: **Output: Text File**
3: Use the relevant PDFBox package to convert pdf file into text file
4: Read the pdf file such as Rama.pdf
5: PDFParser is used to extract the text from the parser
6: Use getText() function to retrieve all the strings from the pdf file
7: Store the extracted strings into text file

Algorithm 4 Algorithm for encryption of text file into image shares

1: **Input: Text file**
2: **Output: Image Shares**
3: Read the text file which contains some secret information
4: Intitialize the 2 random noisy image shares such as Image1, Image2 with png format
5: Move all the information from text file into buffer
6: Calculate the height and width of the image shares
7: Width = No. of Characters in a line
8: Height = No. of lines in a file
9: Read each line from the contents in a buffer
10: begin
11: Select each character from the line
12: Compute ascii value for that character
13: Calculate the individual pixel using SetRGB() in Java
14: Place every pixel of a line in Image1 and Image2 alternately
15: Store pixel on the image shares based on its x, y coordinates which denote the position of the character in a line and line number respectively
16: This process is continued till the end of the file
17: end
18: Finally save the Image shares such as Image1, Image2 in png format using ImageIo.write () in Java

Algorithm 5 Algorithm for converting random noisy images into scrambled images

1: **Input: Image files**
2: **Output: Scrambled Images**
3: Initialize the two scrambled image shares as Image1, Image2 with png format
4: Interchange width and height of the random noisy images as height and width of the scrambled images
5: Read the pixel in each random noisy image share
6: begin
7: Get the pixel value using getRGB method
8: Store that pixel in the corresponding scrambled image using setRGB method
9: This process is repeated till the end of the file
10: end
11: Finally save the scrambled Image shares such as Image1, Image2 in png format

5 SPDFUEVC Decryption Process

Retrieving the original pdf file from the scrambled images stored in the cloud storage using SPDFUEVC decryption technique involves three phases. In the initial phase, the random noisy image shares are extracted from the scrambled images and then they are converted into a text file in the next phase. The image shares are in.png format and every line of them must be read. Then each pixel from every line must be retrieved using getRGB method. Each one of them must be rewritten in hexa code and the resultant value entered in a string buffer. This should be continued till the end of the file. Finally the entire contents in the buffer must be rewritten as a text file and in the last phase, the text file is converted into the original pdf using Apache PDFbox application program interface as shown in Figure 4.

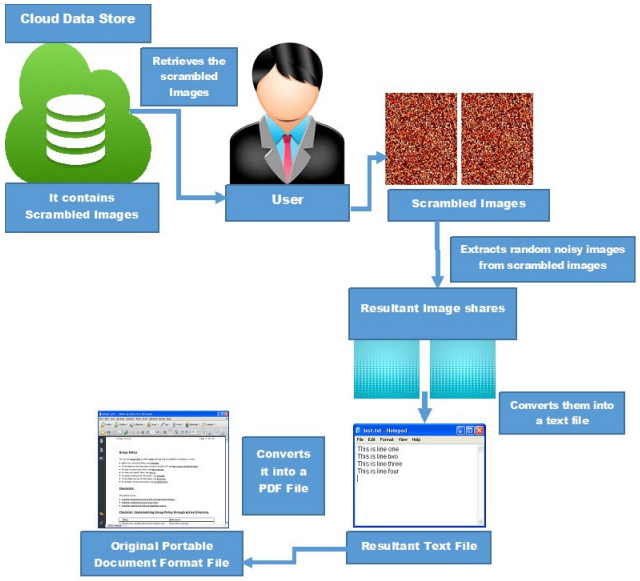

Figure 4: PDF file retrieved from scrambled images

Algorithm 6 Extraction of random noisy images from the scrambled images

1: **Input: Scrambled images**
2: **Output: Random noisy images**
3: Intitialize the two random noisy image shares as Image1, Image2 with png format
4: Interchange width and height of the scrambled images as height and width of the random noisy images
5: Read the pixel in each scrambled image share
6: begin
7: Get the pixel value using getRGB method
8: Store that pixel in the corresponding random noisy image using setRGB method
9: This process is repeated till the end of the file
10: end
11: Finally save the random noisy image shares such as Image1, Image2 in png format

6 Security Analysis

In this section, the security attribute of this technique have been discussed with various factors.

Confidentiality. The proposed technique reveals only the encrypted file information to the CSP and all

Algorithm 7 Algorithm for encryption of Text file into Image shares

1: **Input: Random noisy image files**
2: **Output: Secret text file**
3: Read the random noisy image files
4: Initialize the string buffer
5: Select each and every pixel from the line using getRGB() in Java
6: Find ascii value for the selected pixel
7: Find appropriate character from ascii value and place it in a buffer
8: Rewrite all the data from the buffer into a text file using FileWriter() in Java
9: Finally save the Image files in png format

Algorithm 8 Conversion of a text file into original pdf file

1: **Input: Text file**
2: **Output: pdf file**
3: Use file reader to read the source file
4: Store all the contents to buffered image
5: Create a pdf file using PDFwriter component
6: Use pdfDoc.setMargins() function to set the margins of pdf file
7: Read each and every line from the buffer
8: Set the font size and style with setFontFamily(), setFontSize() function respectively
9: Store the text in the document using setText() function
10: Repeat the process until file comes to an end
11: Finally save the content as a pdf file

other details maintained by TTP (Trusted third party). This ensure that sensitive information protected from CSP and illegal user.

Integrity. The original data files are converted into scrambled random noisy shares and then uploaded into the cloud. The proposed technique ensure that no one can modify the data.

Access Control Management. The DO is revealing the file access control details to the TTP in a secure manner. This avoids the illegal users to modify the access control details.

Prevention of Intruder. The proposed technique prevents the intruder to access the data transferred in the communication channel between the communicating parties such as DO, CSP, User and TTP by enforcing the encrypted form of data transfer.

7 Experimental Result

The above-said SPDFUEVC technique has been evaluated in Java [19, 31] and various tests have been worked out using a laptop with the configuration of 2.40 GHz, Intel core i3 processors with 4GB RAM on Windows 8 Professional version 1. The algorithm is implemented in various sizes of pdf files; the performance of the technique is evaluated with parameters such as execution time and size of the image shares for SPDFUEVC encryption and decryption technique. During the encryption process the original pdf file is converted into a text file then it is converted into random noisy image shares which are then converted into scrambled images. The data confidentiality of this algorithm is compared to that of conventional symmetric encryption algorithms such as AES (Advanced Encryption Standard) and DES (Data Encryption Standard) used in the current cloud domain [1, 40].

The test result of the pdf file for the proposed SPDFUEVC technique is as follows.

7.1 SPDFUEVC Encryption Process

7.1.1 Conversion of the pdf File into a Text File

The pdf file is converted into a text file using apache PDF Box program interface. Figure 5 and Figure 6 show the sample pdf file and the resultant text file after the conversion.

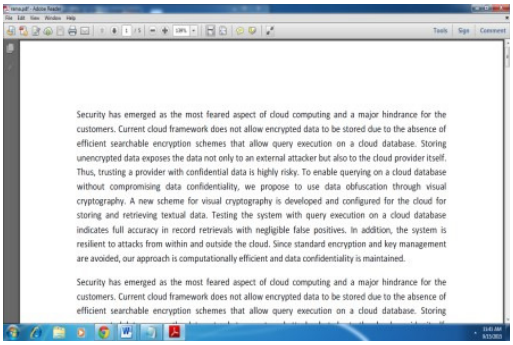

Figure 5: Sample pdf file

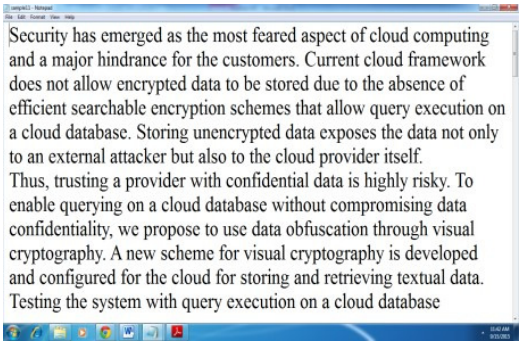

Figure 6: Resultant text file

7.1.2 Conversion of Text File into Image Shares Using SPDUFEVC Encryption Technique

The resultant text file is encrypted into image files using SPDFUEVC encryption technique. Figures 7 and 8 show

the resultant image files after encryption process.

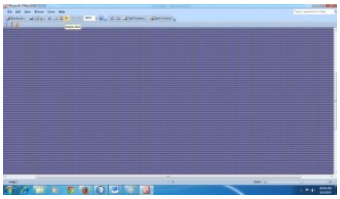

Figure 7: Image1.png (1090 x 10)

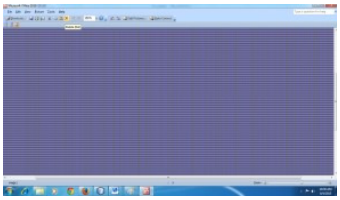

Figure 8: Image2.png (1090 x 10)

7.1.3 Algorithm for Converting Random Noisy Image Shares into Scrambled Images

The resultant image files are converted into scrambled images. Figures 9 and 10 show the random noisy images converted into scrambled images.

Figure 9: Scrambled Image1.png (10 x 1090)

Figure 10: Scrambled Image2.png (10 x 1090)

7.2 SPDFUEVC Decryption Process

7.2.1 Extraction of Random Noisy Images from the Scrambled Images

The random noisy image shares are extracted from scrambled images. Figure 11 and Figure 12 show the random noisy images extracted from scrambled images.

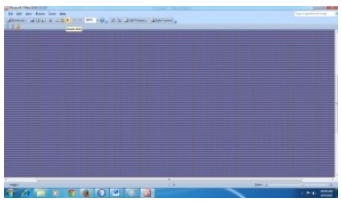

Figure 11: Image1.png (1090 x 10)

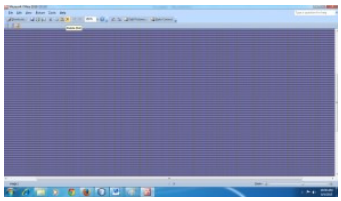

Figure 12: Image2.png (1090 x 10)

7.2.2 Conversion into Text File from the Random Noisy Image Shares

The random noisy image shares are converted into a text file using SPDFUEVC technique. Figure 13 shows the resultant text file after the decryption process.

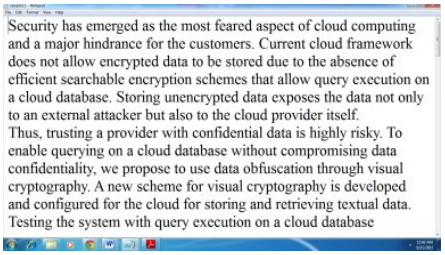
Security has emerged as the most feared aspect of cloud computing and a major hindrance for the customers. Current cloud framework does not allow encrypted data to be stored due to the absence of efficient searchable encryption schemes that allow query execution on a cloud database. Storing unencrypted data exposes the data not only to an external attacker but also to the cloud provider itself.
Thus, trusting a provider with confidential data is highly risky. To enable querying on a cloud database without compromising data confidentiality, we propose to use data obfuscation through visual cryptography. A new scheme for visual cryptography is developed and configured for the cloud for storing and retrieving textual data. Testing the system with query execution on a cloud database

Figure 13: Resultant text file

7.2.3 Conversion of Text File into pdf File

The resultant text file is converted into pdf file using apache PDF Box application programming interface. Figure 14 shows the resultant pdf file after the conversion.

This algorithm has been applied on various pdf files and it is found that the size of the image file is comparatively lesser than that of the original pdf file during the SPDFUEVC encryption algorithm. Table 1 describes the different sizes of pdf files juxtaposed with those of image files and Figure 15 juxtaposes the sizes of the original pdf files with those of image files.

Table 2 juxtaposes the sizes of the original pdf files with those of the deciphered pdf files and Figure 16 discusses the correlation of the sizes of the original pdf files with those of the deciphered pdf files. This analysis proves that both the pdf files are of same size.

7.2.4 Execution Time for Encryption

The time taken for conversion of pdf file into text file, then into image shares and then into scrambled images

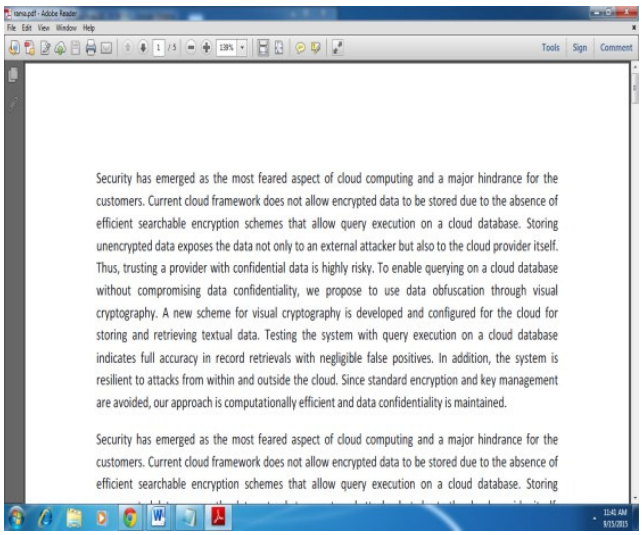

Figure 14: Resultant pdf file

Table 1: Size of Image shares during SPDFUEVC encryption

Size of original pdf files (in KB)	Size of image files (in KB)
6	2
9	2
15	4
21	4
30	4
38	4

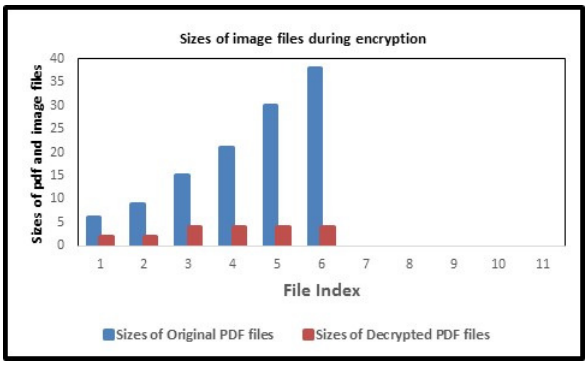

Figure 15: Size of image files during SPDFUEVC encryption

is calculated as execution time in encryption process and also the time for conversion of scrambled images into random noisy image shares, then into text file and finally into original pdf file is considered as execution time in decryption process. This process is compared with conventional DES and AES algorithms which are used in the current cloud domain. Table 3 compares this execution time with that of AES and DES.

The execution time for encryption in SPDFUEVC technique is found to be comparatively lesser than those in DES and AES for various sizes of pdf files. The Figure 17 depicts the time taken for the execution of these algorithms.

Table 2: Comparison of sizes of original and decrypted pdf files

Size of original pdf files (in KB)	Size of deciphered pdf files (in KB)
6	6
9	9
15	15
21	21
30	30
38	38

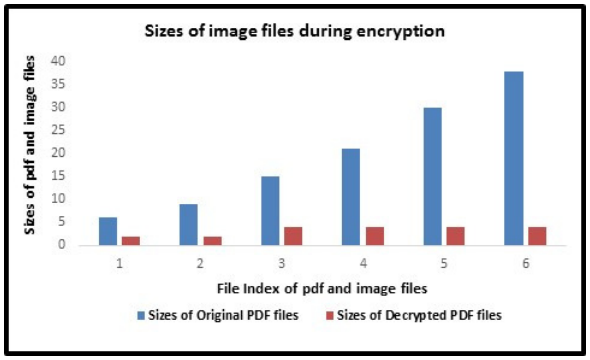

Figure 16: Size of image files during SPDFUEVC decryption

Table 3: Execution time for encryption in SPDFUEVC, AES and DES

Size of Pdf files (in Kb)	Execution time of SPDFUEVC (in ms)	Execution time for AES (in ms)	Execution time for DES(in ms)
6	431	671	531
9	483	702	565
15	655	862	734
21	734	934	804
30	890	1090	950
38	1077	1251	1111

7.3 Execution Time for Decryption

Table 4 compares the execution time for decryption in SPDFUEVC with those of AES and DES. The execution time for decryption in SPDFUEVC technique is found to be marginally lesser than those in DES and AES. The Figure 18 denotes the time taken for the execution of these algorithms.

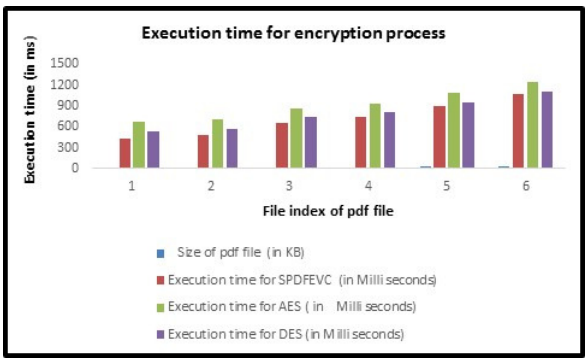

Figure 17: Encryption execution time for SPDFUEVC, AES and DES

Table 4: Execution time for decryption in SPDFUEVC, AES and DES

Size of Pdf files (in Kb)	Execution time of SPDFUEVC (in ms)	Execution time for AES (in ms)	Execution time for DES(in ms)
6	277	282	280
9	284	288	286
15	285	293	289
21	302	306	304
30	309	314	312
38	316	320	318

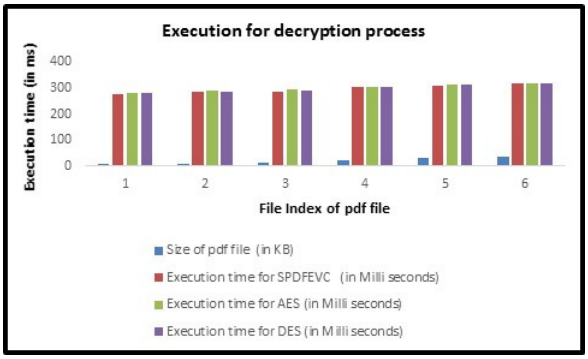

Figure 18: Decryption execution time for SPDFUEVC, AES and DES

8 Complexity

All the previous visual cryptography approaches have been so far used to hide only a small amount of information in an image. In the current scenario, when researchers try to hide a large amount of information in it, they have to face the challenges of increased share size and image processing time and also poor quality of retrieved image resulting in obtaining only optimum solutions [12, 38, 47]. The proposed SPDFUEVC technique can effectively hide a larger amount of textual data with minimum effort, space and time complexity and retrieve the whole information with data confidentiality and integrity.

9 Conclusion

The proposed new technique named (SPDFUEVC) Securing Portable Document Format file Using Extended Visual Cryptography ensures data integrity and confidentiality in the cloud storage. The traditional visual cryptography technique has so far assured confidentiality only to image file. But the proposed approach provides the same for the pdf file using visual cryptography technique and proves to be more efficient than the current cloud storage techniques. The complexity of this approach is shown to be reasonable and it is much less than those of standard algorithms. In this technique, storage entry is fully protected and hence prohibitive to any unauthorized entity. The proposed fool-proof technique ensures data confidentiality and security along with integrity and reputation.

References

[1] Amazon, *Amazon EBS Encryption Now Available*, May 2014. (`https://aws.amazon.com/about-aws/whats-new/2014/05/21/Amazon-EBS-encryption-now-available/`)

[2] S. Abdulla, "New visual cryptography algorithm for colored image," *Journal of Computing*, vol. 2, no. 4, pp. 4–15, 2010.

[3] M. Arunachalam and K. Subramanian, "Aes based multimodal biometric authentication using cryptographic level fusion with fingerprint and finger knuckle print," *International Arab Journal of Information Technology*, vol. 12, no. 5, pp. 431–440, 2015.

[4] G. Ateniese, C. Blundo, A. De Santis, and D. R. Stinson, "Visual cryptography for general access structures," *Information and Computation*, vol. 129, no. 2, pp. 86–106, 1996.

[5] C. Blundo, P. D'Arco, A. De Santis, and D. R. Stinson, "Contrast optimal threshold visual cryptography schemes," *SIAM Journal on Discrete Mathematics*, vol. 16, no. 2, pp. 224–261, 2003.

[6] C. Blundo, A. De Santis, and M. Naor, "Visual cryptography for grey level images," *Information Processing Letters*, vol. 75, no. 6, pp. 255–259, 2000.

[7] C. Blundo, A. De Santis, and D. R. Stinson, "On the contrast in visual cryptography schemes," *Journal of Cryptology*, vol. 12, no. 4, pp. 261–289, 1999.

[8] K. Brindha and N. Jeyanthi, "Secured document sharing using visual cryptography in cloud data storage," *Cybernetics and Information Technologies*, vol. 15, no. 4, pp. 111–123.

[9] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[10] Z. Cao, C. Mao, L. Liu, "Analysis of one secure anti-collusion data sharing scheme for dynamic groups in the cloud," *International Journal of Electronics and Information Engineering*, vol. 5, no. 2, pp. 68–72, 2016.

[11] V. Chang and M. Ramachandran, "Towards achieving data security with the cloud computing adoption framework," *IEEE Transactions on Services Computing*, vol. 9, no. 1, pp. 138–151, 2016.

[12] S. Chen, "A visual cryptography based system for sharing multiple secret images," in *Proceedings of the 7th WSEAS International Conference on Signal Processing*, pp. 113–118, Chicago, Aug. 2007.

[13] C. Esposito, A. Castiglione, and K. K. R. Choo, "Encryption-based solution for data sovereignty in federated clouds," *IEEE Cloud Computing*, vol. 3, no. 1, pp. 12–17, 2016.

[14] J. B. Feng, H. C. Wu, C. S. Tsai, Y. F. Chang, and Y. P. Chu, "Visual secret sharing for multiple secrets," *Pattern Recognition*, vol. 41, no. 12, pp. 3572–3581, 2008.

[15] B. L. Gunjal and S. N. Mali, "Design and implementation of invisible and visible color image watermarkingwith netbeans ide," *International Journal of Computer Applications*, vol. 71, no. 11, pp. 25–45, 2013.

[16] T. Hofmeister, M. Krause, and H. U. Simon, "Contrast-optimal k out of n secret sharing schemes in visual cryptography," *Theoretical Computer Science*, vol. 240, no. 2, pp. 471–485, 2000.

[17] A. E. A. El Hossaini, M. El Aroussi, K. Jamali, S. Mbarki, and M. Wahbi, "A new robust blind copyright protection scheme based on visual cryptography and steerable pyramid," *International Journal of Network Security*, vol. 18, no. 2, pp. 250–262, 2016.

[18] Y. C. Hou, "Visual cryptography for color images," *Pattern Recognition*, vol. 36, no. 7, pp. 1619–1629, 2003.

[19] Java, *Java*, 2015. (`http://www.java2s.com/`)

[20] L. M. Kaufman, "Data security in the world of cloud computing," *IEEE Security & Privacy*, vol. 7, no. 4, pp. 61–64, 2009.

[21] A. Le, A. Markopoulou, and A. G. Dimakis, "Auditing for distributed storage systems," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2182–2195, 2016.

[22] C. C. Lin and W. H. Tsai, "Visual cryptography for gray-level images by dithering techniques," *Pattern Recognition Letters*, vol. 24, no. 1, pp. 349–358, 2003.

[23] C. H. Ling, C. C. Lee, C. C. Yang, and M. S. Hwang, "A secure and efficient one-time password authentication scheme for WSN," *International Journal of Network Security*, vol. 19, no. 2, pp. 177–181, 2017.

[24] C. Liu, R. Ranjan, X. Zhang, C. Yang, D. Georgakopoulos, and J. Chen, "Public auditing for big data storage in cloud computing–a survey," in *IEEE 16th International Conference on Computational Science and Engineering*, pp. 1128–1135, harvard, Dec. 2013.

[25] P. Mell and T. Grance, "Effectively and securely using the cloud computing paradigm," tech. rep., Oct. 2009.

[26] A. Kr. Mishra and A. Gupta, "Visual cryptography for gray scale image using block replacement half toning method," *African Journal of Computing & ICT*, vol. 6, no. 4, pp. 53–58, 2013.

[27] A. Mosa, H. M. El-Bakry, S. M. Abd El-Razek, S. Q. Hasan, "A proposed E-government framework based on cloud service architecture," *International Journal of Electronics and Information Engineering*, vol. 5, no. 2, pp. 93–104, 2016.

[28] M. Naor and A. Shamir, "Visual cryptography," in *Workshop on the Theory and Application of of Cryptographic Techniques*, pp. 1–12, berlin, May 1994.

[29] M. Naor and A. Shamir, "Visual cryptography ii: Improving the contrast via the cover base," in *International Workshop on Security Protocols*, pp. 197–202, berlin, apr. 1996.

[30] P. K. Naskar, H. N. Khan, and A. Chaudhuri, "A key based secure threshold cryptography for secret image," *International Journal of Network Security*, vol. 18, no. 1, pp. 68–81, 2016.

[31] Netbeans, *Netbeans IDE*, 2012. (`https://netbeans.org/`)

[32] N. Ojha and S. Padhye, "Cryptanalysis of multi prime rsa with secret key greater than public key," *International Journal of Network Security*, vol. 16, no. 1, pp. 53–57, 2014.

[33] R. De Prisco and A. De Santis, "Color visual cryptography schemes for black and white secret images," *Theoretical Computer Science*, vol. 5, no. 10, pp. 62–86, 2013.

[34] J. Sandeep and A. Manjeed, "Embedded extended visual cryptography scheme," *Journal of Computer Engineering*, vol. 8, no. 1, pp. 41–47, 2012.

[35] C. E. Shannon, "A mathematical theory of communication," *Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

[36] J. Singh, "Cyber-attacks in cloud computing: A case study," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 78–87, 2014.

[37] W. Teng, G. Yang, Y. Xiang, T. Zhang, and D. Wang, "Attribute-based access control with constant-size ciphertext in cloud computing," *IEEE Transactions on Cloud Computing*, vol. pp, no. 99, pp. 1–1, 2015.

[38] C. C. Thien and J. C. Lin, "Secret image sharing," *Computers & Graphics*, vol. 26, no. 5, pp. 765–770, 2002.

[39] H. Tian, Y. Chen, C. C. Chang, H. Jiang, Y. Huang, Y. Chen, and J. Liu, "Dynamic-hash-table based public auditing for secure cloud storage," *IEEE Transactions on Services Computing*, 2015. (doi: 10.1109/TSC.2015.2512589)

[40] Z. Wan, J. E. Liu, and R. H. Deng, "Hasbe: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing," *IEEE*

Transactions on Information Forensics and Security, vol. 7, no. 2, pp. 743–754, 2012.

[41] B. Wang, M. Li, and H. Wang, "Geometric range search on encrypted spatial data," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 704–719, 2016.

[42] D. Wang, F. Yi, and X. Li, "On general construction for extended visual cryptography schemes," *Pattern Recognition*, vol. 42, no. 11, pp. 3071–3082, 2009.

[43] Z. Wang, Y. Lu, G. Sun, "A policy-based deduplication mechanism for securing cloud storage," *International Journal of Electronics and Information Engineering*, vol. 2, no. 2, pp. 70–79, 2015.

[44] C. Wu and L. Chen, "A study on visual cryptography (masters thesis)," tech. rep., Sept. 1998.

[45] Z. Yan, W. Ding, X. u, H. Zhu, and R. H. Deng, "Deduplication on encrypted big data in cloud," *IEEE Transactions on Big Data*, vol. 2, no. 2, pp. 138–150, 2016.

[46] Z. Yan, M. Wang, Y. Li, and A. V. Vasilakos, "Encrypted data management with deduplication in cloud computing," *IEEE Cloud Computing*, vol. 3, no. 2, pp. 28–35, 2016.

[47] C. N. Yang and T. S. Chen, "Size-adjustable visual secret sharing schemes," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 88, no. 9, pp. 2471–2474, 2005.

Biography

K. Brindha, M. E.(CSE) from Sathayabama University, Chennai, Tamilnadu, India is presently working as Assistant Professor (Selection Grade) in the School of Information Technology and Engineering, VIT University, Vellore. Currently she is doing research work at VIT University, Vellore, Tamil nadu, India. She is the author of several articles published in reputed journals. Her research interest include cryptography, network security and image processing.

Dr. N. Jeyanthi is an Associate Professor in School of Information Technology and Engineering, VIT University, India. She received her Ph.D. and M.Tech. in Information Technology with Networking as specialisation from VIT University, BE in Computer Science and Engineering from Madurai Kamaraj University, India. Her research interest is on network security in real-time applications. She published around thirty international journal papers and many international conference papers. She received the Active Researcher Award from VIT University for three consecutive years. She is an editorial board member of international journals and acted as programme chair in many international conferences. She is a life member of Indian Society of Technical Education. Her research interest includes Cryptography, Network Security, IoT.