ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 9(37), DOI: 10.17485/ijst/2016/v9i37/102107, October 2016* ISSN (Online) : 0974-5645

Compressed and Encrypted Online Multimedia File Storage in Cloud

Saipavan Narasaraj*, Vishnuvardhan Reddy and Vaishnavi Moorthy

Department of Computer Science and Engineering, SRM University, Kattankulathur - 603203, Tamil Nadu, India; Saipavan_narasaraj@srmuniv.edu.in; tottireddy_adinarayanan@srmuniv.edu.in; vaishnavi.m@ktr.srmuniv.ac.in

Abstract

Objective: The objective of this paper is to propose a cloud system which ensures high security for storing and sharing user files in the cloud and efficient utilization of cloud storage space. **Methods:** The different methods that can be implemented to achieve a stable and efficient system was incorporated from literature review. Details regarding the working of various compression and encryption standards were extracted to build the proposed system. **Findings:** The internet is growing day by day and can be easily accessed by everyone and everywhere. This increases the need to store and transfer digital data, which would allow users to access data without the hassle of carrying a storage device. The focus of this paper is to propose a technique which ensures the data uploaded in the cloud is secure and also that the storage space is used efficiently. The proposed paper also explains sharing data in a secured format and also how data can be downloaded with minimum bandwidth, which is useful while operating in places where there is minimal internet speed. **Applications/Improvement:** In today's cloud era confidential records or documents involving financial and management information can be securely shared in compressed format.

Keywords: Compressed File Storage, Decryption, Encryption, File Sharing using Cloud, Lossless Compression, Secured File Storage

1. Introduction

The popularity of cloud storage services is increasing rapidly in an effective way to share and backup files[1]. However, there is a concern regarding security and effective usage of storage space among the cloud service providers.

Most of the cloud service providers don't provide the freedom of encrypting the files with the user defined key[2]. This allows the data stored in the cloud to be venerable to attackers, as the service providers might use similar or same key to encrypt user's files. Another possibility is that, encryption is done by using the user's password, which is not reliable; as the data can be accessed by the third parties. Therefore, the login credentials of a particular user's is compromised.

Simultaneously, storage space in the cloud should be utilized efficiently. Most of the systems today store the files without any compression techniques. Lossless compression[3]will reduce the amount of storage required to store the files in the cloud with data integrity, when compared to Lossy compression[4-8].

It is important to make sure that; sensitive file can be shared in an easy and securedmanner. Sharing data with other users should be done is such a way, that only the intended person can access it. But there is no reliable mechanism which can provide it. Also, the time required to download file in its original format is greater when compared to its download in compressed format. This drawback can be overcome, by providing the user with the option of downloading the files in a compressed format.Once the file is downloaded in the users' system, the user can decompress it, to retrieve the original file without compromising the original data signature.

In[9] reported various services that cloud computing platform provide to the users but, there are not many cloud storage services that concentrate on encrypting the files using keys provided by the users. Secret keys are assigned to provide security for the files stored in cloud[10]. In traditional cloud services, the secret keys

* *Author for correspondence*

for encrypting the files are assigned by an algorithm or may be predefined, but in the proposed system files are encrypted by user defined keys. This provides the user a greater role in the security mechanism. In AES encryption standard, encryption is done using keys of length 16 to 256 bits[11]. This is not feasible as it would be troublesome for users to enter large keys. In order to reduce the storage space occupied by files, there is a need to compress them; lossless compression technique will reduce the file size without affecting its original identity with regards to size and quality[12,13]. This feature can be incorporated to the uploaded files and further can be used to reduce the size of the file which needs to be downloaded. The compressed downloaded file can be decompressed using open source software like WINRAR. Symmetrical encryption and compression of files[14] is one among the best ways how a cloud system should as it is effective and efficient. Lossy compression technique will reduce the file size and quality, it is observed, that there is a difference between the lossy compressed multimedia file when compared to its original file[15]. The proposed system will make sure that the compressed file will be the original file when decompressed.

2. Proposed Methodology

This paper intends to present a system which overcomes the above stated drawbacks.The system architecture of the proposed cloud storage service works is shown in Figure1.

File upload: The proposed system enables user to select weather the file should be stored in a high secure manner or in the traditional way. By uploading the file in a normal way, the file will be encrypted with a predefined key; this option can be used while uploading files that are not confidential. On the other hand confidential or important files can be uploaded by the user by specifying the encryption id and key. The uploaded files are encrypted with the key specified by the user. This allows different files of the same user to be encrypted with different keys. If the key of a particular file is compromised, this method ensures security as decryption is only done when the id and key are matched. The presence of encryption id makes it difficult for attackers to access files through brute force as the time required to match two fields is difficult.

Sharing files: File sharing is an important part of cloud service, as it allows quick and easy way of sharing data. The cloud system must make sure that only authorized can access the files. The proposed system allows users to share their files by entering the recipients email id. Users can also enter decryption id and key in the description field.

The contents of the mail include a link to download the file shared along with the description entered by the user. Once the receiver user clicks on the link the user will be directed to a page for verification. The verification page will request the user to enter any three random digits of the sender's phone number. Once the verification has been done, the user is asked enter the decryption id and key. If the user has successfully entered the id and key, the user can now download the file. While sharing a normal file the recipient can directly download the files without any verification.

File download: Once the user selects files and enters the proper verification details if present, the user can download the files. The user is given two options whether to download the files in its original form or to download it in a compressed form. Lossless compression technique is used here, so that the files can retain their original format and size while de compressing it. Downloading compressed file will result in low internet bandwidth usage and also helps faster download.

Modify security information: Users can modify the encryption id of any files uploaded, this feature makes it difficult for attackers to guess the id and key as it can be modified by the user at any time. The encryption key cannot be changed as the contents of the files are encrypted with it and the same key is required to decrypt the file. Users can also modify their phone number; this field is used for verification during file sharing.

Encryption standard: The system uses RSA Encryption standard to encrypt the data.

It is an asymmetric cryptographic algorithm. Asymmetric means that there are two different keys: public and private key. This is also called public key cryptography, because one of them can be given to everyone. The other key must be kept private. The public key is shared by the user to the recipient who will use these details while downloading the file. The algorithm is based on the fact that it is far more difficult to factor a product of two primes than it is to multiply the two primes. The reason for choosing RSA over other encryption standard like AES is due to the reason that, AES encryption is done by using keys that are large in size[16]. For example in AES 128 bit encryption the key contains 13 characters[17,18] ; users find it difficult to enter such large keys and remember

them, as the same key must be utilized during decryption. The size of the encrypted file is larger when compared to the original file. So, it is important to reduce the storage space required for the encrypted files in cloud.

Compression technique: The system uses lossless compression technique to compress the files; this means that the original file can be retained while decompressing the files without change in file quality or size. The technique used here is GZIP[19,20]. Compression takes place using deflate algorithm and decompression takes place using inflate algorithm.

Deflate algorithm: This algorithm finds duplicated strings presenting the input data. The second occurrence of a string is replaced by a pointer to the previous string, in the form of a pair (distance, length). Distances are limited to 32K bytes, and lengths are limited to 258 bytes. When there is no occurrence of the string in the previous32K bytes, it is emitted as a sequence of literal bytes.Literals or match lengths are compressed with one Huffman tree, and match distances are compressed with another tree. The trees are stored in a compact form at the start of each block. The blocks can have any size (except that the memory available must be able to allocate the compressed data). A block is terminated when deflate function determines that it would be useful to start another block with fresh trees[21].

Inflate Algorithm: Inflate function sets up a first level table that covers some number of bits of input less that the length of longest code. It gets that many bits from the stream, and looks it up in the table. The table will indicate if the next code is that many bits or less and how many, and if it is, it will tell the value, else it will point to the next level table for which inflate function grabs more bits and tries to decode a longer code. How many bits to make the first lookup is a tradeoff between the time it takes to decode and the time it takes to build the table. If building the table took no, then there would only be a first level table to cover all the way to the longest code. However, building the table takes a lot longer for more bits since short codes are replicated many times in such a table. What inflate function does is simply to make the number of bits in the first table a variable, and set it for the maximum speed.

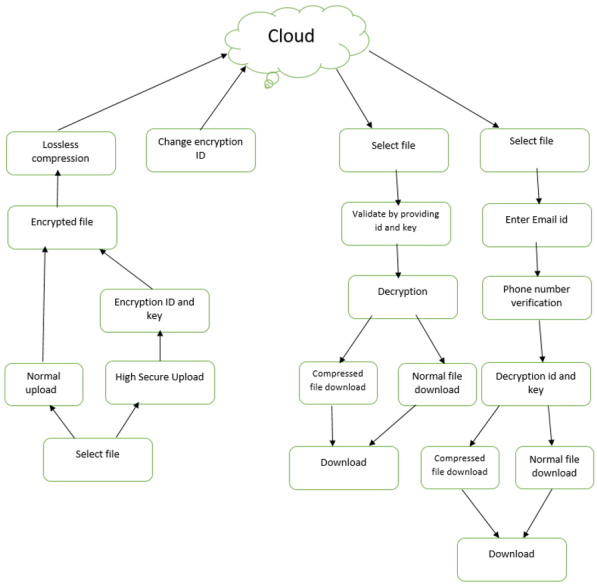

Figure 1. System architecture comprising encryption and compression.

2.1 System Architecture

If a file needs to be uploaded in the cloud through proposed system then there are two methods, one is a normal upload where the file is directly encrypted and another is a highly secure upload where an encryption ID and encryption key are user defined. At the penultimate stage the encrypted file is compressed to reduce the file size[22-25], compression which takes place here is lossless. Finally compressed encrypted file is uploaded into cloud[26,27].

The file that needs to be downloaded can be downloaded by two different methods. One method, which is based on the validation of their ID and key and the other method is through providing their phone number for verification.

In the initial process the file to be downloaded after its validation with user defined identification and key now under goes decryption with the same algorithm as when produced during encryption. At this stage another two options are provided wherein the file can be downloaded as a normal file or a compressed file.

In the second method of obtaining the file from the

cloud, the selected file asks for the validation from the user in terms of a registered e mail id and phone number. Later in this stage the decryption id and key which has been provided during the process of uploading the file. The download of this file, like the prior case again happens to be in two different methods- one , a normal file download and another a compressed file download which aims at reducing the amount of space used by the file

3. Encryption and Compression

When a 270KB file is uploaded the file is encrypted and the file size increases to 729KB, later the encrypted file is compressed to 400KB by using deflate algorithm. This shows that the encrypted by 45%. Later when the file needs to be downloaded the compressed file is decompressed, decrypted to obtain the original file. Along with the original file there is compressed version of the same file[28]. When the user chooses to download the compressed file the zipped file size is 245KB which provides 9.25% compression.

Uploading a multimedia file like an video of size 5.4MB the encrypted file will have a size of 13.8 MB later when the file is compressed its size is reduced to 9.7 MB, here there is 29.7% compression. While downloading the file the compressed file size is 4.9 MB.

Table 1 shows the compression percentage of different file formats. Lossless compression takes place in real time. Text files have the highest compression percentage averaging to above 50 percent. Multimedia files like pictures (gif, jpg, png,etc), audio file such as (mp3, wma, etc) and video files like (avi, mpg, etc) do not exhibit high compression as they are already in compressed format. The amount of compression depends on the presence of redundancy data in the file.Higher the presence of data redundancy, greater is the compression percentage.

Figure2. Gives an idea about how the uploaded and the downloaded file looks like. Here the file uploaded once the file is uploaded in the cloud, the user can download the same file in compressed format. After the file has been downloaded, open source decompression software can be used to obtain the original file. It is inferred from the figure 2, that the downloaded file is exactly similar to the uploaded without any noise in the image.

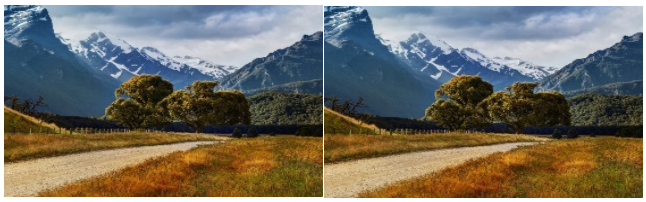

Figure 2. Original file before Downloaded file after Encryption and compression decryption and decompression [280KB].

4. Conclusion

The proposed system enables users to store files in a secure manner. Files can also be shared without the fear of it being compromised. The ability of the files being encrypted with the key entered by the users enables different files being encrypted with different key. Lossless compression is used to compress files; the compression percentage depends on the presence of redundant data in the file.

5. Future Enhancement

To ensure at most level of security the system can be integrated with OTP(one time password)service which ensures that only the user will be able to download his/her files. The way this feature works is, when the user selects the files to download an OTP will be sent to the

Table 1. Comparison between the file sizes at various stages of system operation.

File type	Original file size	Compressed Original file size	Encrypted file size	Compressed encrypted file size	Compression percentage
Text	5.62 KB	2.64 KB	15 KB	8.58 KB	53.02
Document	80.9 KB	75.8 KB	216 KB	148 KB	6.3
PDF	320 KB	297 KB	856 KB	582 KB	7.1
Image	280 KB	252 KB	747 KB	516 KB	10
Audio	1.85 MB	1.7 MB	4.92 MB	3.4 MB	8.1
Video	1.1 MB	1.05 MB	2.94 MB	2.06 MB	4.5

user's phone number, on successfully verifying the users identity the files can be downloaded. This feature can also be extended while sharing files. The user can enter the phone number of the recipient while sharing the file in the cloud system. One the recipient receives and clicks the link to file an OTP will be sent to the recipient's phone number. After validating the recipient the file can be downloaded.

The present encryption technique used in the proposed system increases the file size to a significant amount. Newer encryption technique, which ensures minimal increase in file size, can be utilized to save the cloud storage, without compromising security.

6. References

1. Chachapara K, Bhadlawala S. Secure sharing with cryptography in cloud computing. Nirma University International Conference on Engineering. 2013 Nov. p. 1–3.
2. Chatterjee D, Dasgupta S, Nath J, Nath A. A new Symmetric key Cryptography Algorithm using extended MSA method. International Conference on Communication Systems and Network Technologies. 2011 Jun. p. 89–94.
3. Porwal S, Chaudhary Y, Joshi J, Jain M. Data Compression Methodologies for Lossless Data and Comparison between Algorithms. International Journal of Engineering Science and Innovative Technology. 2013 Mar; 2(2):142–7.
4. Singh P, Suri P. Multimedia Data Compression Techniques. International Journal of Advanced Research in Computer Science and Software Engineering. 2013 Oct; 3(10):321–5.
5. Sethi G, Shaw S, Vinutha K, Chakravorty C. Data Compression Techniques . International Journal of Computer Science and Information Technologies. 2014 Jun; 5(4):5584–6.
6. Blelloch E. Introduction to Data Compression. In: Other Lossy Transform Code. Carnegie Mellon University publ.: USA. 2011; 50–5.
7. Vijayvargiya G, Silakari S, Pandey R. Various Techniques of Image Compression. International Journal of Computer Science and Information Security. 2013 Oct; 11(10):1–5.
8. Razzaque A, Thakur NV. Image compression and encryption: an overview. International Journal of Engineering Research & Technology. 2012 Jul; 1(5):1–7.
9. Kiruba karamoorthi R, Arivazhagan D, Helen D. Analysis of Cloud Computing Technology. Indian Journal of Science and Technology. 2015 Sep; 8(21):1–3.
10. Yanez-Sierra J, Diaz-Perez A, Sosa-Sosa V, Gonzalez JL. Towards Secure and Dependable Cloud Storage Based on User-Defined Workflows. IEEE 2nd International Conference on Cyber Security and Cloud Computing. 2015 Nov. p. 405–10.
11. Parthasarathy MB, Srinivasan B. Increased Security in Image Cryptography using Wavelet Transforms. Indian Journal of Science and Technology. 2015 Jun; 8(12):1–8.
12. Suresh Babu P, Sathappan S. Efficient Lossless Image Compression using Modified Hierarchical Prediction and Context Adaptive Coding. Indian Journal of Science and Technology. 2015 Dec; 8(34):1–6.
13. Kumar S, Bhadauria SS, Gupta R. A Temporal Database Compression with Differential Method. International Journal of Computer Applications. 2012 Jun; 6(48):65–8.
14. Golla T, Klein R. Real-time Point Cloud Compression. IEEE/RSJ International Conference on Intelligent Robots and Systems. 2015 oct. p. 5087–92.
15. Isenburg M, Matrin. Laszip: lossless compression of lidar data. In: Photogram- metric Engineering and Remote Sensing. American Society publ.: USA. 2013; 209–28.
16. Golla T, Schwartz C, Klein R. Towards efficient online compression of incrementally acquired point clouds. The Eurographics Association. 2014 Oct; 17–22.
17. Chaudhari M, Saxena K. Fast and Secure Data Transmission using Symmetric Encryption and Lossless Compression. International Journal of Computer Science and Mobile Computing. 2013 Feb; 2(2):58–63.
18. WEP Key Generator. Available from: http://www.andrews-companies.com/tools/wep.asp. Date accessed: 23/05/2016.
19. Data Compression. Available from: https://en.wikipedia.org/wiki/Data_compression. Date Accessed: 19/05/16.
20. Huang Y, Peng J, Kuo C, Gopi M. A generic scheme for progressive point cloud coding. Institute of Electrical and Electronics Engineers. 2008 Mar-Apr; 14(2):440–53.
21. Pahal R, Kumar V. Efficient Implementation of AES. Advanced Research in Computer Science and Software Engineering. 2013 Jul; 3(7):1–6.
22. Alfalou A, Brosseau C, Abdallah N, Jridi M. Simultaneous fusion, compression and encryption of multiple images. Optics express. 2011 Nov; 19(24):24023–9.
23. Subhamastan Rao T, Soujanya M, Hemalatha T, Revathi T. Simultaneous Data Compression and Encryption. International Journal of Computer Science and Information Technologies. 2011 Jun; 2(5):2369–74.
24. Ito M, Ohnishi N, Alfalou A, Mansour A. New image encryption and compression method based on independent component analysis. IEEE Information and Communication Technologies From Theory to Application. 2008 Apr; 1–6.
25. Cheng H, Li X. Partial encryption of compressed images and videos. IEEE Transactions On Signal Processing. 2000 Aug; 48(8):2439–51.
26. Pan D. A tutorial on MPEG/Audio compression. Institute of Electrical and Electronics Engineers [IEEE] Multimedia. 1995 Jun; 2(2):60–74.
27. Atoum MS, Abdulgader O, Al- Rababah, Al-Attili A. Network Security on cloud computing. International Journal of Computer Science And Network Security. 2011 May; 11(5):1–5.
28. Jose S. Storage and Retrieval of images from Video Databases. SPIE Workshop CA. USA. 1995 Feb.