Hybrid Compressed Hash Based Homomorphic ABEncryption Algorithm for Security of data in the Cloud Environment

Pachipala Yellamma[1,*]
Research Scholar, Dept of computer science, Bharathiar University, Coimbatore, Tamilnadu,India.
pachipala.yamuna@gmail.com

Dr. Challa Narasimham[2]
Professor & Principal, Dept of CSE, Vignan's Institute of Information Technology, Vishakhapatnam, Duvada, Andhra Pradesh, India
narasimham_c @yahoo.com

P Yellamma[3]
Asst. Professor, Dept of CSE, K L University, Vaddeswaram, Guntur, Andhra Pradesh, India.
pachipala.yamuna@kluniversity.in

*Corresponding author *pachipala.yamuna@gmail.com*

***Abstract*:-** Cloud computing is an emerging technology in the world of computing. It provides a convenient virtual environment for on-demand access to different type of services and computing resources such as applications, networks and storage space in an efficient way. The virtual environment is a massive compound structure in terms of accessibility that made easy in a compact way and familiar of functional components. The complexity in virtual environment generates several issues related to data storage, data security, authorization and authentication in cloud computing. With the size of the data, it becomes difficult to the cloud user to store large amounts of information in the remote cloud servers due to high computational cost, insecurity and costs high per hour proportional to the volume of information. In this paper, we propose compressed hash based encrypted model for the virtual environment. The aim of this paper is to store huge amount of data in the cloud environment in the form of compressed and encrypted data in a secure way.

Keywords: *ABEncryption, Chaotic-map, Compression, DCT-DWT, Decryption, Dynamic Byte-stream, Encryption, Haar-wavelet, Hash model, Homomorphic-encryption.*

*****

I. INTRODUCTION

Cloud computing provides services like delivering applications that include allocation of computer processing power, network, and storages on demand, in a pay as you go mode. While Users store an enormous amount of personal confidential data, chances for exposing of valuable data and privacy issues exists. Cloud applications are running somewhere in the cloud model. Cloud User can retrieve any amount of data, at anytime from anywhere on the web. Generally, a user doesn't know the exact location of their data and services were closed. The process of keeping user applications and data in a secured way is the first place such as education, healthcare, governance, transportation systems, energy systems and mobile communication. Key security considerations for cloud environments is to contain, Authentication, Authorization, Securing data at rest, Securing data in motion, Identity and Access management, Data Integrity and Key management. This paper includes Authentication, Authorization and Data protection as some of the security considerations for cloud computing [1].Authentication refers to specifying the entity requesting access to some protected data. The authentication policies are always under the control of the organizations and altered at their own convenience [2]. Authorization delegates specifying the access rights to the protected resources using access policies.

The massive size multimedia data is remarkably increased on the cloud. Compressing the huge amount of data leads to the good use of cloud storage. Different types of compression techniques are available to compress the textual and multimedia data. Data compression is the method of transforming, encoding, and changing the bits structure of data, such that it absorbs a smaller amount of space on the disk. Based on the reconstruction compression could be classified into two ways, one is Lossless compression technique and the other is Lossy compression technique. Lossless compression scheme implicates no information loss. Comparatively, this technique suits the best for text information [3]. It is extremely significant for the reconstruction which is as same as the original information. Whereas Lossy compression schemes involve some loss of information which is acceptable based on application. Information compressed by using Lossy schemes normally cannot be reconstructed accurately. This technique is best suitable for the images, audio, and video data.

The efficiency of Lossless and Lossy data compression algorithm can measure depending on the ratio and size of compression as well as the time and speed of processing and entropy [4].This paper also focuses on what amount of compression has to done for given different applications. An approach for measuring compression ratio is the bytes before compression and to bytes after compression [5].The compression ratio can calculate as.

$$compression\ Ratio = {}^{A_1}/_{A_0} * 100$$

A_1=Number of bytes after compression.

A_0=Number of bytes before compression.

After that, compression data can upload to cloud in a secure way. Securing information in the cloud is crucial for cloud computing applications wherein the data flows from applications to storage and storage to applications. There are different types of threats that can occur to the data in the cloud like replay attacks, man-in-middle attacks, denial of service and unauthorized access. Encryption is the process of transforming plaintext phase to a scrambled phase. Decryption transforms data from a scrambled phase to plaintext [6]. Encryption exists in two different types one is symmetric encryption, the other is asymmetric encryption. The confidential key is in use for Symmetric encryption and decrypting the data/information. Symmetric encryption can unsafe because the confidential key has to swap between the parties and anyone who manages to get the secret key could decrypt the data. Asymmetric encryption has two different types of keys, one is the public key and the other is the private key [7]. The two keys are interlinks such that one key encrypts plaintext to the coded text and the other key decrypts coded text to plaintext [8]. Public key acknowledges to all users and the private key is confidential. Asymmetric encryption is best suited for securing data between the parties.

Hash function input is an arbitrary message M; compute a stable length Hash code H(M), which sometimes is called a message digest or hash value or hash code h=H(M). The hash function can use outside the security but not all the hash functions are used for security [6]. The cryptographic hash functions that we are concerned with have different types of functions such as MD families (like MD3, MD4, and MD5) and SHA (Secure Hash Algorithm) family like SHA1, SHA2 (bits are 224,256,384,512 [9]. MD5 popular messages digest produces a 128-bit length hash values and as well collisions can be found in 2^{21}hashes. SHA1 produces a 160-bit hash value, but it does not offer any better security where collisions can found in 2^{61}hashes. MD5 and SHA1 hash functions itself are not considered secret or secured [10].SHA and MD families have defects, including security problems, hardware functioning and high computational have complications in cloud computing [11].

To deal with these limitations, this paper proposes a light-weight method is an Extended Quadratic chaotic map. The proposed system not only provides a wider range of data with low computational cost but also a well-organized system compared to other existing systems. This cryptosystem provides different pseudorandom byte streams for each session. The Extended Quadratic chaotic map allows for Text, Images, audio and video files (multimedia files) of any size.

II. Related Work

In cloud computing, storage is a type of service model in which information is managed, maintained, store, available and access remotely in virtualized environment with the help of client software to the client specifies, backup set and then data transfer across the internet.

A. Data compression

Multiple numbers of organizations are heading towards cloud computing environment for storing a huge amount of file/data. Data storage is the essential service for the cloud; based on space and time so many data compression techniques are available to produce the data in a compact form in the virtual environment. Different Lossless and Lossy data compression techniques are proposed and used in this paper.

TABLE I: SHOWS PREDICTIVE ANALYSIS OF DIFFERENT LOSSLESS COMPRESSION TECHNIQUES WITH VARIOUS PARAMETERS.

predictive analysis of different lossless compression techniques			
Parameters	***Run length***	***Huffman***	***LZW***
Compression Ratio	Good	Average	Poor
Compression time	Less	Average	More
Decompression time	Less	More	Moderate
Compressed file size	Poor	Average	Good
Compressed pattern matching	No	Yes	No
Permits Random access	No	Yes	No

a). Huffman Encoding

Huffman encoding algorithm was implemented by Huffman David in 1952 at MIT [3]. Huffman algorithm works basing on the chance of all the letters of the file to calculate the frequency distribution. Accordingly, basing on the chance of the symbol code-words assigned. Lengthier codeword's for smaller probabilities and shorter code words for higher probabilities [3]. Huffman code words can determine by creating a binary tree consecutively; which can encode by building it from top to bottom. Huffman coding is a more successful technique for text and video compression, analyzing the frequency of amount of symbols or pixels in the data [12]. In this paper, Huffman coding algorithm is applicable to text data (.Doc, .Docx, .ppt and partly video files) in cloud computing. It is suitable for images like MPEG and JPEG with a combination of other algorithms. In lossless compression, the output is an exact replica of the input image when the file uncompressed [3]. Discrete Cosine Transform commonly abbreviated as DCT is a technique used to convert spatial domain to frequency components/domain. To compress an image we remove a measure of information to reach compaction in the cloud. But care should be taken to which the information has to discard to get compressed. Due to computational efficiency, DCT is popularly used as it works on the principle [13]. This paper is to provide complete video compression using a hybrid method of DCT-DWT with optimal Huffman coding that gives more compression ratio [14] and less compression time. Haar Wavelet Technique is well-known for being simple and fast.HWT is best suitable for the audio and image compression. It involves in two ways, forward and reverses transforms [15].

b).Forward Transform

Computations of scaling coefficients-add two adjacent sample values and divided by 2, and computation of wavelet coefficients- subtract two adjacent samples values and divided by 2.

c).Inverse transforms

Computation requires simple addition and subtraction. Consider two neighbouring samples p and q, so forward transform can be achieved by;

$Average(A) = (p+q)/2$

And $Difference(D) = (p-q)/2$

The inverse transform is applied to get the original sample values.

$p = (A-D)/2$

And $q = (A+D)/2$

Simple steps to calculating HT is

step1: Calculating the average of each pair of samples from the array of values (N/2).
step2: Calculating the difference between the average value and sample value of an array.
step3: Set the primary half with averages and the next half with differences.
step4: Reiterate this process until the first half.

B. Hash function

Chaotic hash function is a nonlinear dynamic system, which accomplishes the necessary characteristics for the cryptosystem. The Chaotic-key hash function is constructed based on the MD model (Merkel &Damgard) that takes an arbitrary length of message M and produces fixed length of message n. Input message M is split into a predetermined length of blocks m. M= {M_1, M2, M3.....M_m} and apply the padded message to the input, it produces the output message making its length to multiples of n and to determine the length of the message it is based on the last n=2 bits [9]. This process is iterated until the last block. Finally, the last block of a padded message contains the total length of the binary code.

C. Encryption and Decryption

These days data security is a very important issue in the communication environment. Many techniques, algorithms, protocols and methods are available for solving the security [16]. Each and every algorithm and technique is having its own pros and cons. In the virtual environment (cloud) data security is the major issue [17]. Cloud-based encryption algorithms are available for securing the data. The encryption process is done by using different existing encryption algorithms [18].

In public-key encryption (PKE), the message can be encrypted for the particular receiver based on receiver's public key [19]. Then as advancement, the Identity-Based Encryption (IBE) is introduced, which replaces the public key by an arbitrary length of the string. Attribute-Based-Encryption is relatively same as PKE. In attribute-Based-Encryption, a user key and secret message are considered by a set of expressive attributes and specific keys are able to decrypt a specific secret message [20]. ABE have mainly two approaches: CPABE and KPABE (cipher text-policy & key-policy attribute based encryption) [20]. In CPABE, access policy is associated with the cipher-text whereas KPABE, access policy is associated with private-key; each scrambled message is joining together through a set of attributes. This paper proposes a Dynamic-key policy ABE in which Data security is the major consideration in cloud computing. This algorithm generates hash values based on hash function.

III. PROPOSED MODEL

We propose a model called Compressed Quadratic hash based Homomorphic ABEncryption for the virtual environment. In this paper, we propose the algorithms for Data compression, key Generation, and Data encryption algorithms and then implement these algorithms for the virtual environment. Apply the different file formats like .doc, .docx, .ppt, .xls, images, audio and video inputs of the proposed compressed models; the output of the compressed data is the input of the proposed hash model. Proposed hash model generates hash values called hash key. The input of proposed encryption algorithm is compressed data and hash key which generates encrypted secured data. This proposed model is to reduce the storage space and place files in a secured form. Data is stored in the unreadable format, so that unauthorized users will not be able to access the data. Only authorized user can access the data by using the key credentials and policies. This model proposes more compression ratio and more secure than the other models.

In figure 1, Authenticated super-user is uploading the file. After the uploading, the file is compressed based on specified compression algorithm and generates a hash key based on the Extended Quadratic chaotic-map algorithm. Hash-based Homomorphism ABEncryption takes two inputs for encryption in which the file one is a compressed file and the other is hash key. The encrypted file is successfully uploaded into the AWS console Bucket. The Authorized super-user receives the Hash key to decrypt the file.
This paper proposes three algorithms. They are
Algorithm 1: Hybrid compression algorithm,
Algorithm 2: Extended Quadratic Chaotic-Map Algorithm,
Algorithm 3: Hash-based Homomorphic ABEncryption Algorithm

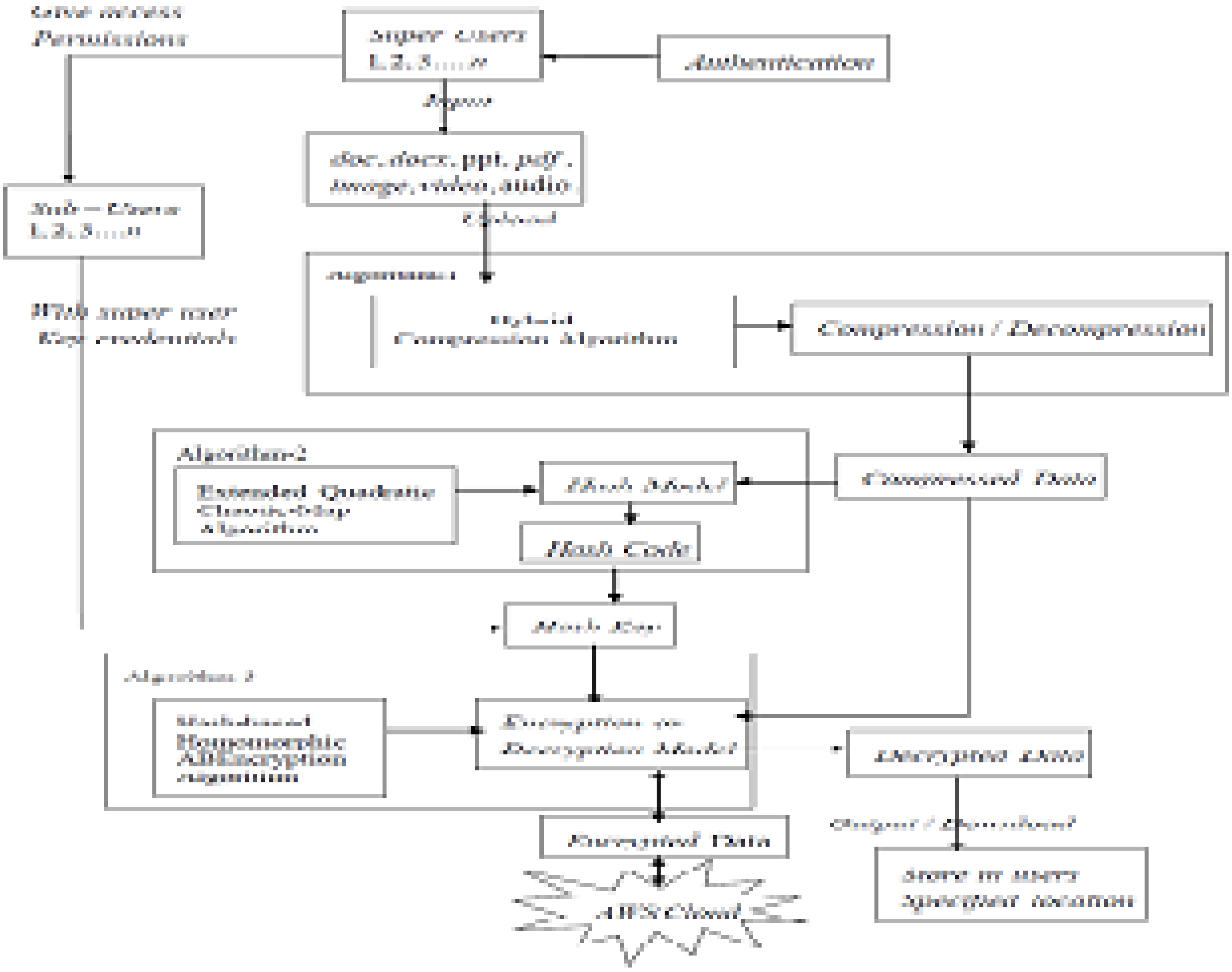

Give access Permissions
Super Users 1,2,3.....n
Authentication
Input
doc, docx, ppt, pdf, image, video, audio
Upload
Sub – Users 1,2,3.....n
Algorithm-1
Hybrid Compression Algorithm
Compression / Decompression
With super user Key credentials
Algorithm-2
Extended Quadratic Chaotic-Map Algorithm
Hash Model
Hash Code
Compressed Data
Hash Key
Algorithm 3
Hash-based Homomorphic ABEncryption algorithm
Encryption or Decryption Model
Decrypted Data
Output / Download
Encrypted Data
Store in users specified location
AWS Cloud

Figure 1: Compressed Hash Based Encryption Model for the virtual environment.

A. Pseudo code of Compressed Hash Based Encryption model for the virtual environment.

```
Input:  Doc, Docx, pdf, Video and Audio Files
Output: Compressed hash based encrypted file
Procedure:
Step 1: Input user specific file
Step 2: Connect the cloud server using the AWS key pair.
Step 3:  Proposed Compression models
 Step 3a: if file .extension=".doc"   or ".Docx"   or ".ppt" or "pdf"   then
          Apply Dynamic Byte stream Based Huffman File compression algorithm
          Apply Extended Quadratic Chaotic-Map Algorithm
          Apply Hash-based Homomorphic ABEncryption Model
   Else if file=="image file"              then
Step 3b: Fimage=Filter image file using Variance median filter
Convert the Fimage file into 8x8 blocks.
  Apply Dynamic Byte stream Based Haar DWT wavelet compression
  Apply Extended Quadratic Chaotic-Map Algorithm
  Apply Hash-based Homomorphic ABEncryption Model
        Else if file =="video file"       then
End if Else
 Convert input file into byte or binary stream
 Apply Dynamic Byte stream Based DWT-DCT model
       Apply Extended Quadratic Chaotic-Map Algorithm
            Apply Hash-based Homomorphic ABEncryption Model
Step 3c: Frames [] =extract frames (file);
For each frame in Frames []         Do
Convert frame[i] to byte data YCbCr
Apply Dynamic Byte stream Based DWT-DCT compression
     Apply Extended Quadratic Chaotic-Map Algorithm
Apply Hash-based Homomorphic ABEncryption Model
Done
Else if file="audio file"   then
 Step 3d: Convert audio file to streaming byte data.
  Apply Dynamic Byte stream Based Haar wavelet algorithm
      Apply Extended Quadratic Chaotic-Map Algorithm
 Apply Hash-based Homomorphic ABEncryption Algorithm
```

The above algorithm is for the solution of Compressed Hash Based Encryption model for the virtual environment.

This paper proposes the different algorithms for the Compressed Hash-based Homomorphic ABEncryption are given below.

a). Proposed Dynamic byte stream file compression algorithms.

Dynamic Byte stream Based Huffman File compression algorithm for text files.

1) Input file as F
2) If F=='Doc' || F=='Docx'|| F=='Pdf'|| F=='Ppt'
3) then
4) Transform F into byte stream array.
5) Read the byte stream from the input file F.
6) Partition the F into blocks of size 8-bits
7) Sort the byte stream in the descending order of their values.
8) Select the highest mode byte-sets.
9) For each byte in byte-sets
10) Do
11) Compute the entropy, average length and redundancy factors on the byte-sets using equation (1),(2),(3).

$$Entropy = \sum (P_K .(\log_2 P_K))......(1)$$

$$AvgLength = \sum (P_K \times L_K)........(2)$$

$$\text{Redundancy} = \left(AvgLength - Entropy / Entropy\right) \times 100.....(3)$$ 12).

Set MSB=1 for the selected bit of the byte sets Whose probability and computing values are higher than given threshold.

13). Set MSB=0 for the non-selected bit of the byte sets whose probability and computing values are lower than given threshold.

14). done

15). Else if F=="Image"

b). Dynamic Byte stream Based Haar DWT wavelet compression for image files

1) Read RGB image
2) Extract R,G,B components
3) Create Haar Matrix HW with 8x8 size as

$$\mathbf{W} = \begin{bmatrix} 1 & 1 & 2 & 0 & 4 & 0 & 0 & 0 \\ 1 & 1 & 2 & 0 & -4 & 0 & 0 & 0 \\ 1 & 1 & -2 & 0 & 0 & 4 & 0 & 0 \\ 1 & 1 & -2 & 0 & 0 & -4 & 0 & 0 \\ 1 & -1 & 0 & 2 & 0 & 0 & 4 & 0 \\ 1 & -1 & 0 & 2 & 0 & 0 & -4 & 0 \\ 1 & -1 & 0 & -2 & 0 & 0 & 0 & 4 \\ 1 & -1 & 0 & -2 & 0 & 0 & 0 & -4 \end{bmatrix}$$

4) Partition each component into 8 block size and represented as X.
5) Apply DWT on each block using the following equation

$$HW = W \cdot X \cdot W'$$

6) Create mask with 8x8 block size as

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

7) Multiply M to each component as Haar wavelet Result $(HWR) = M \cdot X$

8) Decompress the image using Inverse DWT on each HWR using

$$HW' = W' \cdot HWR \cdot W$$

16).else if F==’audio’

Dynamic Byte stream Based Haar wavelet compression for audio files same as of image files. Convert audio files to streaming byte data.

17). Else if F==’Video’

c). Dynamic Byte stream Based DWT-DCT compression for video files.

a) Input video byte data in YCbCr format
b) Partition data into 16x16 blocks
c) Apply 1st level 2D DWT on data
d) Apply 2D DWT on LL sub-block
e) Apply 2D-DCT using proposed
f) Compress 4x4 blocks.

Repeat steps (c) to (f) to each block

$$Y = 0.299 \times R + 0.587 \times 0.115 \times B$$
$$Cb = -0.169 \times R - 0.331 \times G + 0.5 \times B$$
$$Cr = 0.5 \times R - 0.418 \times G - 0.082 \times B$$

These algorithms are repeated for various file formats up to the last byte. Output of the compressed file is input of the proposed hash model.

B. Proposed Extended Quadratic Chaotic-Map Algorithm.

An Extended Quadratic Chaotic-Map Algorithm is a new keyed hashing scheme based on a single chaotic map.

Algorithm

1) Initialize input compressed data as H
2) The given input message H is padded so that the last n=10 bits of the final block in the padded message have binary representation of the entire length of the message.
3) The input message M is initially padded with a bit of 0x, and an enough number of 0s to get a message M_0 so that $|H'|= n/2 \pmod n$
4) The binary representation of |H| (in n/2 bits) is then appended to the end of H' to obtain H''
5) This padded message H'' is splits into m blocks (H[1]; H[2]; H[3]; ... H[m]), each of length n.
6) H'' = H[1]+ H[2]+H[3]+ ... +H[m],
7) Done

This Extended Quadratic Chaotic-Map Algorithm is used for generation of hash-key value.

C. Proposed Hash-based Homomorphism ABEncryption Model.

a).Setup

This phase is used to setup the public key parameters of the user using the mathematical group theory functions.

$$Public\ key := \{g(p), g(q), g(r), G_{\alpha 1}, G_{\alpha 2}, G_{\alpha 3}, H^1_{AK}, H^2_{SK}, H^3_{policies}\}$$

b).Key Generation

This phase generates the private key as a secret key, depending on the set of user’s credentials and Hash value as attributes. Each user is associated with secret key and it will be generated using three pattern keys as

$$Public\ key := \{g(p), g(q), g(r), G_{\alpha 1}, G_{\alpha 2}, G_{\alpha 3}, H^1_{AK}, H^2_{SK}, H^3_{policies}\}$$

$Master\ key := \{\alpha 1, \alpha 2, \alpha 3\}$ Are taken randomly from cyclic group

$K_{1,i} = g_p^{1/(S'+\alpha 1)}$; i=0…..Partition1.length;

$K_{1,j} = g_q^{1/(S'+\alpha 2)}$; j=0….. Partition2.length;

$K_{1,k} = g_r^{1/(S'+\alpha 3)}$; k=0…. Partition3.lenght;

$$Secret\ key := \{Hashvalue.length; H^1_{AK}, H^2_{SK}, H^3_{policies}; K_{1,i}, K_{1,j}, K_{1,k}\}$$

c).Cloud Data Encryption

Each cloud user encrypts the data using his public key along with the credentials and Homomorphic encryption.

d).Cloud Data Decryption

This phase enables a receiver with the matching credentials to decrypt the cloud data using Homomorphic decryption.

Here, compressed data is taken as input for encryption. Additive Homomorphic and multiplicative Homomorphic are performed on two bytes of the compressed data. This process is repeated to all the compressed byte stream of the file. If users’ attribute set $S \notin T$ next result is ⊥. Elsewhere the algorithm chooses T elements from *K*. The Hash-based Homomorphic

ABEncryption Algorithm for encryption and decryption is represented as

e).Homomorphic Encryption

$$\mathrm{Enc}(\mathrm{M}_1 + \mathrm{M}_2) := \mathrm{Enc}(\mathrm{C}_0 + \mathrm{C}_0^{'})$$

$$:= Enc\,(C_0) + Enc\,(C_0^{1})$$

$$:= (C_0 + \gamma * \beta)\; mod\; n^2 + (C_0^{'} + \gamma * \beta)\; mod\; n^2$$

$$\mathrm{Enc}(\mathrm{M}_1 \cdot \mathrm{M}_2) := \mathrm{Enc}(\mathrm{C}_0 \cdot \mathrm{C}_0^{'})$$

$$:= Enc\,(C_0) \cdot Enc\,(C_0^{1})$$

$$:= (C_0 + \gamma + \beta)\; mod\; n^2 + (C_0^{'} + \gamma + \beta)\; mod\; n^2$$

f). Homomorphic Decryption

$$\mathrm{Dec}(\mathrm{Enc}(\mathrm{M}_1 + \mathrm{M}_2)) := (\,\mathrm{Enc}(\mathrm{C}_0 + \mathrm{C}_0^{'}))\ \mathrm{mod}\ \alpha$$

$$:= (C_0 + \gamma * \beta)\; mod\; n^2 + (C_0^{'} + \gamma * \beta)\; mod\; n^2$$

$$:= C_0 + C_0^{'}$$

$$\mathrm{Dec}(\mathrm{Enc}(\mathrm{M}_1 \cdot \mathrm{M}_2)) := (\,\mathrm{Enc}(\mathrm{C}_0 \cdot \mathrm{C}_0^{'}))\ \mathrm{mod}\ \alpha$$

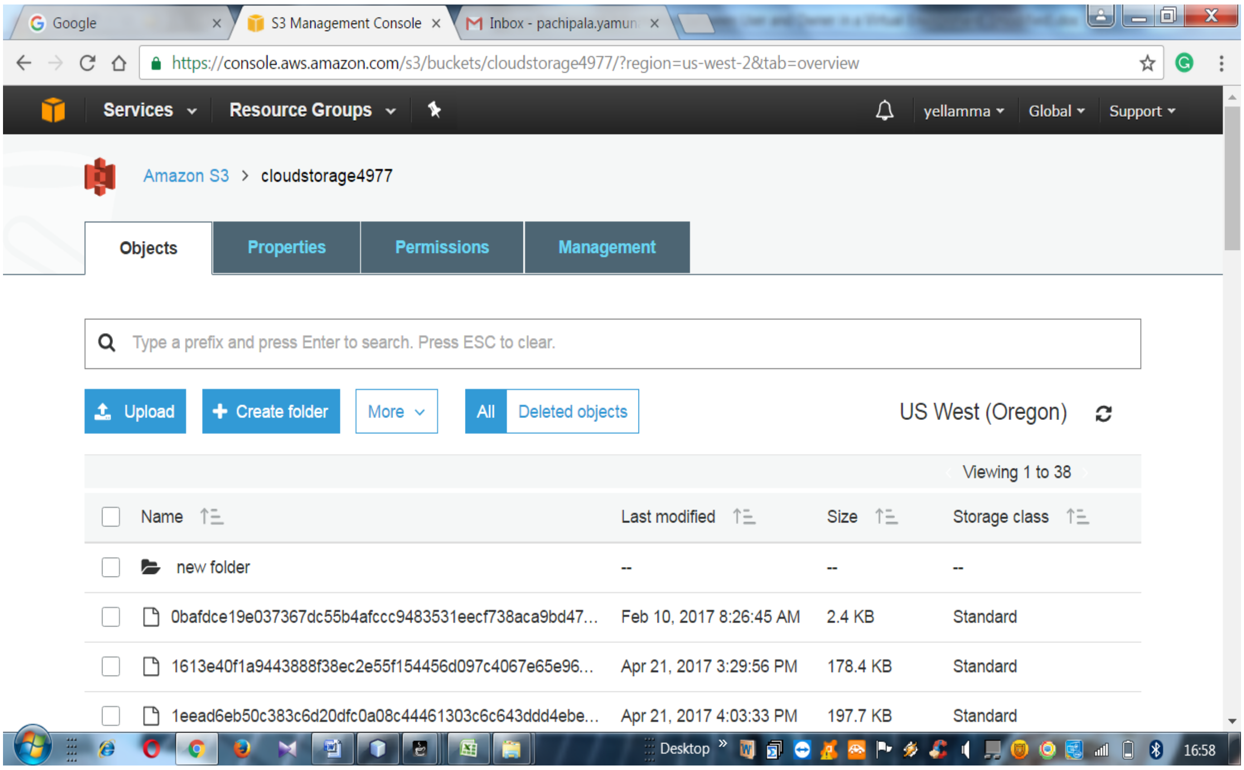
$$:= (C_0 + \gamma + \beta)\; mod\; n^2 + (C_0^{'} + \gamma + \beta)\; mod\; n^2$$

$$:= C_0 \cdot C_0^{'}$$

IV. RESULTS AND DISCUSSION

This paper is implemented using AWS console for the cloud environment.AWS bucket contains objects of encrypted files stored in the cloudstorage4977 bucket.

Different super users have different file states and graphs; it contains the super-user name, super-user Email, sub-user Email, Filename, FileSize, FileCompressionSize, and FileCompresRatio. Individual super-user uploaded a different file into the cloud. Their Original file size (Bytes) is represented in file states and graphs. If an unauthorized user needs the file, he/she needs to get the permission from the authorized super user.



Name	Last modified	Size	Storage class
new folder	--	--	--
0bafdce19e037367dc55b4afccc9483531eecf738aca9bd47...	Feb 10, 2017 8:26:45 AM	2.4 KB	Standard
1613e40f1a9443888f38ec2e55f154456d097c4067e65e96...	Apr 21, 2017 3:29:56 PM	178.4 KB	Standard
1eead6eb50c383c6d20dfc0a08c44461303c6c643ddd4ebe...	Apr 21, 2017 4:03:33 PM	197.7 KB	Standard

Figure 2: Compressed and hash based Encrypted files are stored in Amazon S3 contain cloudstorage4977 bucket.

The Authorized super user gives permission with his/her key credentials. Unauthorized user register is based on super user key credentials. Now, an unauthorized user is called sub-user of a particular super user. Sub-user downloads/decrypts the file based on the hash key of a particular file. Sub-user can have the access permission of specified file; not possible to download/decrypt other files. The super-user is authorized to delete the sub-user permission. This paper contains multiple numbers of super users and sub-users.

TABLE II: DIFFERENT FILE FORMATS ARE UPLOADED INTO THE VIRTUAL ENVIRONMENT USING COMPRESSED HASH-BASED HOMOMORPHIC ABENCRYPTION MODEL.

Different file formats are uploaded into the virtual environment		
Name of the file	***Original file size in bytes***	***Compressed file size in bytes***
MIJS-template-2016.doc	605184	57420
running.docx	1237448	1226105
Babucarrtoons.pdf	9321737	8080288
Bruce Lee-belike water.mp4	9135335	9096306
economics.mp3	6062391	5998583
Chal- Chal- Gurram - 3D Animation.mp4	4658550	4643153
These aurelienwailly.pdf	5700461	4241095
sample-file-sic-modif-fin.doc	234496	182646 182646

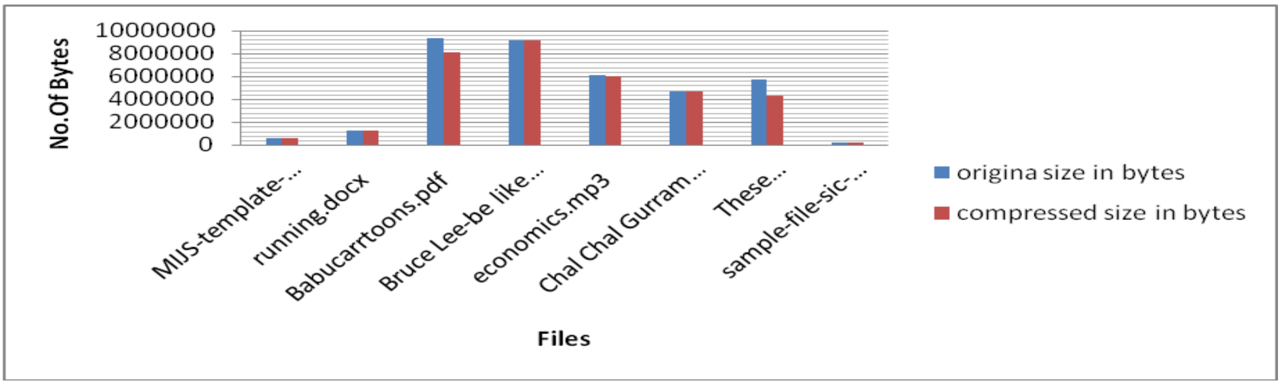


Figure 3: An individual super-user uploaded different files into the cloud. Their Original file size (Bytes) and compressed file size (Bytes) is represented in file states. Using compressed hash-based Homomorphic ABEncryption model.

V. CONCLUSION

Cloud computing is the environment which provides convenient and on-demand access to different types of services and computing resources such as applications, servers, and networks in an efficient way. The main issue in the cloud computing is how to store, secure and access user's data into the cloud. In this paper, we resolved these issues using a novel compression and integrity based encryption and decryption model on different file formats. We have implemented novel compression models, hash model and hash key based Homomorphic encryption and decryption model in the cloud environment. Experimental results show that the proposed models have high less storage and less communication cost compared to traditional cloud security models. In future, this work can be extended to the cloud-based Hadoop environment on big data.

REFERENCES

[1] Kawser Wazed Nafi[1], Tonny Shekha Kar[2], Sayed Anisul Hoque[3], Dr. M. M. A Hashem[4] ('2012). A Newer User Authentication, File encryption and Distributed Server Based Cloud Computing security architecture, International Journal of Advanced Computer Science and Applications, Vol. 3, No. 10.

[2] Pachipala Yellamma et al," the survey of an efficient search scheme over encrypted data on mobile cloud Tees", International Journal of Pure and Applied Mathematics, Volume 117 No. 19 2017, 379-382,ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)

[3] P.Yellamma, Dr.Challa Narasimham (2012). Performance Analysis of Different Data Compression Techniques on Text File, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 1 Issue 8.

[4] Neha Sikka, Sanjay Singla (2016).Lossless Image Compression Technique using Haar Wavelet and Vector Transform, International Conference on Research Advances in Integrated Navigation Systems(RAINS-2016), April 06-07,2016-IEEE.

[5] Dalvir Kaur1, Kamaljeet Kaur2 (2013).Analysis of Lossless Data Compression Techniques, International Journal of Computational Engineering Research||Vol, 03||Issue, 4|| April||2013-pp-123-127.

[6] Rajdeep Bhanot1 and Rahul Hans2 (2015). A Review and Comparative Analysis of Various Encryption Algorithms, International Journal of Security and Its Applications, Vol. 9, No. 4, pp. 289-306.

[7] Pachipala Yellamma et al," Data Security for Cloud Using Public Key Cryptosystem", IJCTA, 9(10), 2016, pp. 4545-4552, International Science Press

[8] Mayank Patwal and Tanushri Mittaly (2014).A Survey of Cryptographic based Security Algorithms for Cloud Computing, HCTL Open Int. J. of Technology Innovations and Research HCTL Open IJTIR, Volume 8,e-ISSN: 2321-1814, ISBN (Print): 978-1-62951-499-4.

[9] A.Kanso, M.Ghebleh (2013).A fast and efficient chaos-based keyed hash function, Commun Nonlinear Sci Numer Simulat 18, Elsevier, 109-123

[10] Kirti Aggarwal, Dr. Harsh K. Verma (2015). Hash_RC6 - Variable Length Hash Algorithm using RC6, 2015 International Conference on Advances in Computer Engineering and Applications (ICACEA),IEEE.

[11] Yantao Li (2016). Collision analysis and improvement of a hash function based on chaotic tent map, Optik 127(2016) 4484-4489.

[12] T. Bernatin, G. Sundari (2014).Video Compression Based on Hybrid Transform And Quantization with Huffman Coding for Video Codec, International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT) 978-1-4799-4190-2/14, IEEE-2014, pp-452-456.

[13] Parekar P.M., Thakare S.S (2014).Lossless Data Compression Algorithm-A Review, International Journal of Computer Science and Information Technologies, vol.5 (1), 276-278.

[14] X.Y. Wang, D.D. Zhang (2014).Discrete wavelet transform-based simple range classification strategies for fractal image coding, Nonlinear Dyn,75, 3, pp. 439–448.

[15] Ranu Gupta (2014). Image Compression using Haar Wavelet Transform and chaos-Based Encryption, IJCSI International Journal of Computer Science Issues,Vol.11,Issue 2,No 1,ISSN(Print):1694-0814 | ISSN(Online):1694-0784.

[16] Pachipala Yellamma et al.." Intelligent Data Security in Cloud Computing",International Journal of Current Engineering and Technology E-ISSN 2277 – 4106, P-ISSN 2347 – 5161, February 2014, Vol.4, No.1

[17]

[18] B.Nithya, P.sripriya (2016). Comparative Analysis of Symmetric Cryptographic Algorithms on.Net Platform, Indian Journal of science &Technology, volume 9, Issue 27.

[19] Harsh Yadav, Mayank Dave (2014).Secure Data Storage Operations with Verifiable Outsourced Decryption for Mobile Cloud Computing, IEEE International conference on Recent Advances and Innovations in Engineering (ICRAIE-2014) , Jaipur, India.

[20] Pachipala Yellamma ",data security in cloud using RSA", 4th ICCCNT – 13 July 4 - 6, 2013, Tiruchengode, India,IEEE-31661

[21] Baodong Qin, Robert H.Deng, Shengli Liu, and Siqi Ma (2015).Attribute-Based Encryption with Efficient Verifiable Outsourced Decryption, IEEE Transactions on Information Forensics and Security, ISSN(c) 1556-6013.