International Journal of Advance Research in Engineering, Science & Technology(IJAREST),
ISSN(O):2393-9877, ISSN(P): 2394-2444,
Volume 2,Issue 4, April- 2015

Secure File Sharing Using Single To Multicloud

Kaveri Marathe [1], Priyanka Chavan [2], Savita Latpate[3], Prof.Y.B.Gurav [4]

[1] Computer, PVPIT, kmarathe15@gmail.com
[2] Computer,PVPIT,priya.yogesh.toke@gmail.com
[3]Computer,PVPIT,savitalatpate09@gmail.com
[4]Computer,PVPIT,ybgurav@gmail.com

Abstract

In recent years use of Cloud computing in different mode like cloud storage, cloud hosting, cloud servers are increased in industries and other organization as per requirements. While considering the power, stability and the security of cloud one can't ignore different threats to user's data on cloud storage. Data access control is an effective way to ensure the data security in the cloud. However, due to data outsourcing and untrusted cloud servers, the data access control becomes a challenging issue in cloud storage systems. Existing access control schemes are no longer applicable to cloud storage systems, because they either produce multiple encrypted copies of the same data or require a fully trusted cloud server. Malicious user at cloud storage is become most difficult attacks to stop. In proposed system we are implementing the concept of multiple server along with enhanced security using encryption techniques where rather storing complete file on single cloud system will split the file in different chunks then encrypt and store it on different server and the meta data required for decrypting and rearranging a file will be stored in metadata management server.

Keywords- *Cloud, Multicloud, Encryption, Decryption, Splitting, Merging, Service providers.*

I. INTRODUCTION

The existing data storage system allows the data storage on a single server at the same location. The user system provides password protection scheme for data access. With the rapid development and application of cloud computing and cloud data storage is taking place, users are more concerned about the security and privacy issues involved with their private data stored on the different server .This has resulted in the need of having an efficient data storage and security mechanism for the Cloud System.

The cloud computing era has resulted in many significant changes in the field of data storage. The current data storage poses a risk to the security of the user's data as the user's personal data is stored on centralized data storage. From the users point of view this is an eminent threat to his data on the server as the probability of an access by unauthorized user is increased. It is suggested that the data be partitioned into blocks of data to be stored on different data storage units. This security mechanism ensures a reliable data storage .In addition the mechanism ensures integrity of the partitioned data blocks, such that a minimum 'n' number of blocks are required for the successful recovery of the complete file from the CSP.

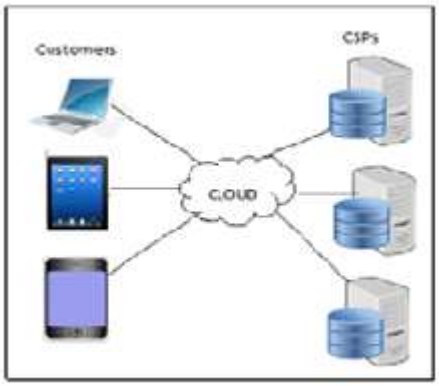

Figure 1. Cloud computing architecture example

A security mechanism for multiple server using file division and encryption technique in cloud computing which provides users with secure data storage and efficient data distribution. The proposed model provides a better alternative to data storage for the user**.** In this paper, we propose a secured cost-effective multi-cloud storage (SCMCS) model in cloud computing which holds an economical distribution of data among the available SPs in the market, to provide customers with data availability as well as secure storage.

II. RELATED WORK

1. To provide secure data storage techniques for files.
2. To secure data from attackers by file division and encryption mechanism.
3. To provide secure encryption key using RSA algorithm.
4. To avoid modification of the user data using data integrity.

Privacy preservation and data integrity are two of the most critical security issues related to user data[4] . In conventional paradigm, the organizations had the physical possession of their data, and thus have an ease of implementing better data security policies. But in case of cloud computing, the data is stored on an autonomous business party that provides data storage as a subscription service. The users have to trust the cloud service provider (SP) with security of their data.

Many of the cryptographic approaches have been proposed for hiding the data from the storage provider and hence preserving data privacy [9] [10] [5]. One bigger concern that arises in such schemes of cloud storage services is that, there is no full-proof way to be certain that the service provider does not retains the user data, even after the user otps out of the subscription. With enormous amount of time, such data can be decrypted and meaningful information can be retrieved and user privacy can easily be reached.

In this paper it is observed that, from a customer's point of view, relying upon a solo Service Provider for his private data is not very secure. In addition, providing better privacy as well as to ensure data availability, it can be achieved by dividing the user's data block into data pieces and distributing them among the available service providers in such a way that no less than a finite number of service providers can take part in successful retrieval of the whole data block. This paper, propose a secured cost-effective multi-cloud storage (SCMCS) model in cloud computing which holds an appropriate distribution of data among the available service providers in the market, to provide customers with data availability as well as secure storage.

III. EXISTING SYSTEM

In existing system when we sending any type of data that data is not secure. Because user can not encrypt that data. So its not secure in a network.That's why when data is flowing in network that time hackers can hack the data easily. So its not secure. So we have developed this system.

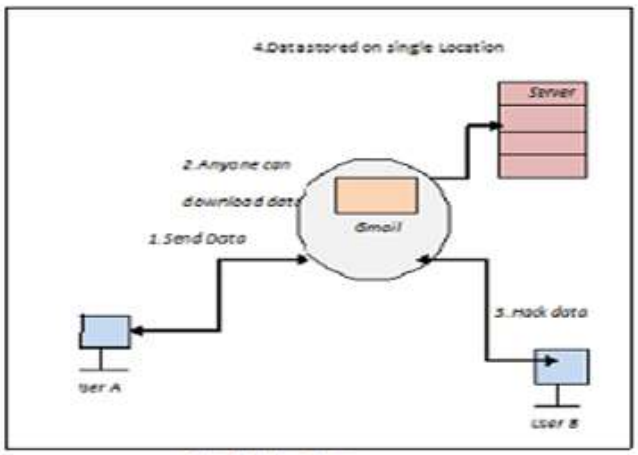

Figure 2.Existing System

IV. PROPOSED SYSTEM

proposed system we are implementing the concept of multiple cloud storage along with enhanced security using encryption techniques where rather storing complete file on single cloud system will split the file in different chunks then encrypt and store it on different server and the meta data required for decrypting and rearranging a file will be stored in metadata management server.

In this proposed system data sharing has done using Cryptography. We have developed one software for Encryption and decryption. Using that system we can do encryption and decryption. When we encrypt the file, this file is split into different parts and stored on different server.

hats why This software is very secure because hackers cannot hack the whole dada. If hackers try to hack data that time they are not able to hack meaningful info.Beacuse that data is not stored on single server. Encrypted data is split into different server.

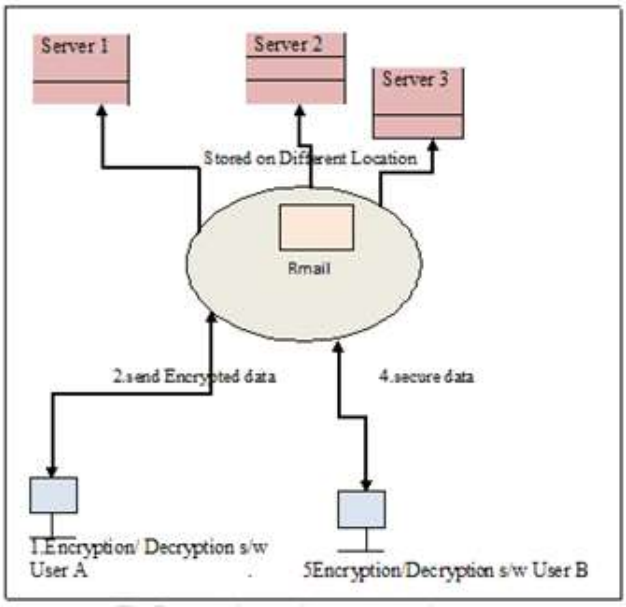

Figure 3 .Proposed System :Secure data sharing

V. ARCHITECTURE

An architectural overview of Secure File Sharing is illustrated in figure.

User Authentication :- In order to use service of uploaded text files from unauthorized user by providing facility of selection of password. proposed system user must be authenticated. This feature allows user to protect access to their

Uploading of File :- This feature provides facility to user to upload their file on the server. Authenticated user can upload as many files as they want and of any type.

Downloading of File :- This feature provides facility to user to download their file on the server. Authenticated user can download as many files as they want and of any type

File Division: This feature is provided by master system(Application). This provides master the ability to divide Text file accepted from authenticated user. This feature is completely controlled by master system only. It means that any other system except master cannot interfere in this process.

Encryption of Text File:This feature provides facility to master store text file on the server in encrypted format. Authenticated user can upload as many text files as they want.Encrption can be done using either AES or blowfish algorithm.

File Decryption: This feature is used by master system when authenticated user wants to access their text file. Master will decrypt the parts of the text file. This feature is provided only for master so that other systems will not able to access this data, if other system gets this data then it's of not use to it as it will be in encrypted format.

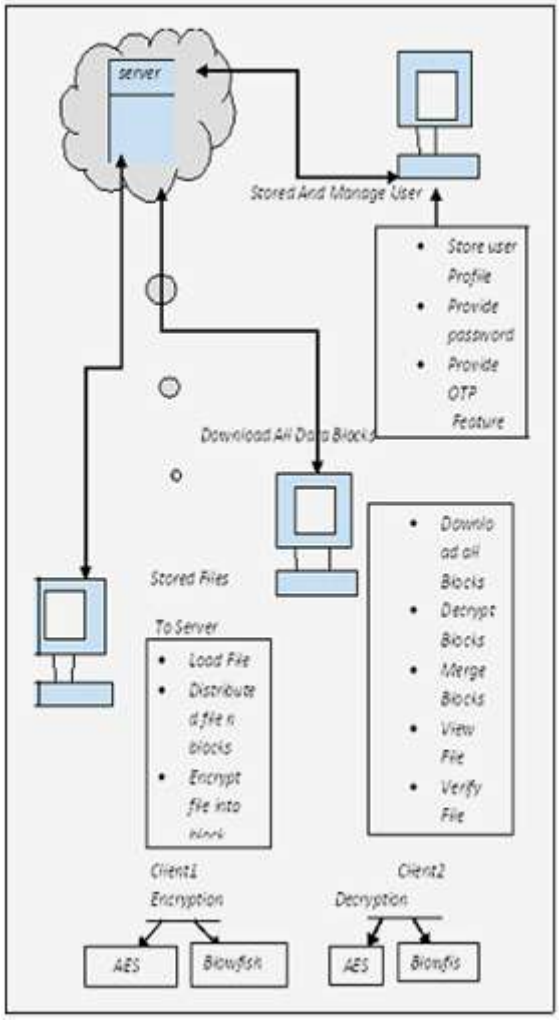

Figure4 .Architecture Diagram

Recombination of Text File parts: This feature is used by master system to recover the original file from the parts of file. File Recombination is one of the most important functions

that are performed by master system to provide accessibility to user to their uploaded file.

VI. DESIGN AND IMPLEMENTATION

6.1 AES

As first publicly accessible, from the NSA for the classification "top secret" approved cipher, the Advanced Encryption Standard (AES) is one of the most frequently used and most secure encryption algorithms available today. Its story of success started 1997, when the National Institute of Standards and Technology NIST announced the search for a successor to the aging encryption standard DES.[2] An algorithm named "Rijndael", developed by the Belgian cryptographists Daemen and Rijmen, excelled in security as well as in performance and flexibility. It came out on top of several competitors, and was officially announced as the new encryption standard AES in 2001. The algorithm is based on several substitutions, permutations and linear transformations, each executed on data blocks of 16 byte – therefore the term block cipher. Those operations are repeated several times, called “rounds”. During each round, a unique round key is calculated out of the encryption key, and incorporated in the calculations. Based on this block structure of AES, the change of a single bit either in the key, or in the plaintext block results in a completely different cipher text block – a clear advantage over traditional stream ciphers. The difference between AES-128, AES-192 and AES-256 finally is the length of the key: 128, 192 or 256 bit – all drastic improvements compared to the 56 bit key of DES. By way of illustration: Cracking a 128 bit AES key with a state-of-the-art supercomputer would take longer than the presumed age of the universe. And Boxcryptor even uses 256 bit keys! As of today, no practicable attack against AES exists. Therefore, AES remains the preferred encryption standard for governments, banks and high security systems around the world.

6.2 The Blowfish algorithm

Blowfish is a symmetric encryption algorithm, meaning that it uses the same secret key to both encrypt and decrypt messages. Blowfish is also a block cipher, meaning that it divides a message up into fixed length blocks during encryption and decryption.[1] The block length for Blowfish is 64 bits; messages that aren't a multiple of eight bytes in size must be padded. Blowfish is public domain, and was designed by Bruce Schneier expressly for use in performance-constrained environments such as embedded systems. It has been extensively analyzed and deemed "reasonably secure" by the cryptographic community.

6.3 RSA

RSA is one of the most successful, asymmetric encryption systems today. Originally discovered 1973 by the British intelligence agency GCHQ, it received the classification “top secret”. Its civil rediscovery is owned to the cryptologists Rivest, Shamir and Adleman, who discovered it during an attempt to break another cryptographic problem. As opposed to traditional, symmetric encryption systems, RSA works with two different keys: A “public” key, and a “private” one. Both work complementary to each other, a message encrypted with one of them can only be decrypted by its counterpart. [3][6]Since the private key can’t be calculated from the public key, the latter is generally made available to the public. Those properties enable asymmetric cryptosystems to be used in a wide array of functions, such as digital signatures. In the process of signing a document, a fingerprint, encrypted with RSA, is appended to the file, and enables the receiver to verify both the sender and the integrity of the document. The security of RSA itself is mainly based on the mathematical problem of integer factorization. A message that is about to be encrypted is treated as one large number. When encrypting the message, it is raised to the power of the key, and divided with remainder by a fixed product of two primes. By repeating the process with the other key, the plaintext can be retrieved back. The best, currently known method to break the encryption requires factorizing the product used in the division. Currently, it is not possible to calculate these factors for numbers greater than 768 bits. None the less, modern cryptosystems use a minimum key length of 3072 bits.

VII. METHODOLOGY

7.1 Registration

In registration get username, email address, password, user generate random verification code (say cd) are asNewRandom.next (0/9). NewRandom.Next ():-function can generate character its six time generate and concatenate the character then execute .we get 6 digit random code. Check existing email address by executing select query where email address is equal to provided value by user otherwise get zero). First login verification code ask status =deactive insert into table all fields with status as deactive and confirmation cod= cd. Send mail to user email address by using SMTP mail class from .NET use parameter for mail:-sender, receiver ,username ,pass message , subject, message body. Create folder with userID as folder name using directory.createdirectory(path) function.(act as personal folder to user) and first part of file will be stored here.

7.2 Login

In this form get userID and password from user execute. *from table where userID=”given by user” and password=”given by user” If this query execute return data , it means user name and password is valid. For this we use SQLconnection.class to establish connection with database.Next we use command to execute command given sql query . This command will return which will connect SqlDataReader.

SqlDataReader.Read: Method is used to read the records. Get user status from this query If(DataReader(“status”)=”deactive”) Then redirect to verification code else redirect to user home. While redirecting to user home, store current userID in session variable. This will allow application to know which user is logged in.

7.3 Verification

Now in this step get the logged username from session variable. If session variable is blank then redirect to login page else execute sql query and confirmation of logged in user from table (a). Get confirmation code on mobile or Email. Now compare a with b, if matched then update user status field with text active and redirect user to the home page.

7.3 Encryption

When user sends any file from his system then that file is first encrypted by using AES(Advanced Encryption Standard) or Blowfish algorithm. The file is encrypted by using any encryption algorithm and the encryption key is generated. But it is necessary to convert this secret key into string format. For that, String com.base64.encodeToSring () function is used.

7.4 Splitting

Once the file is sent from the user system, that file is splits into different parts on different servers. So that , the unauthorized person can’t hack all the data at one time. For this splitting we use readAndFragment() method . This method splits whole file into small fragments.

7.5 Key Encryption

When the encryption key is generated, it is necessary to provide total security to that key. If it is not happened then that key is also hacked by hacker.[7].So it is necessary to encrypt the key. We use RSA algorithm for that purpose. Therefore sufficient security is obtained.

7.6 Decryption

Get the private key i.e. decryption key from key stored .In key stored private key is already stored, but that private key have password and that password is only know to receiver. Using that password receiver can encrypt the private key and after that using that key receiver will able to decrypt the data.[7].Base64.decode () this method is used for decrypt data.

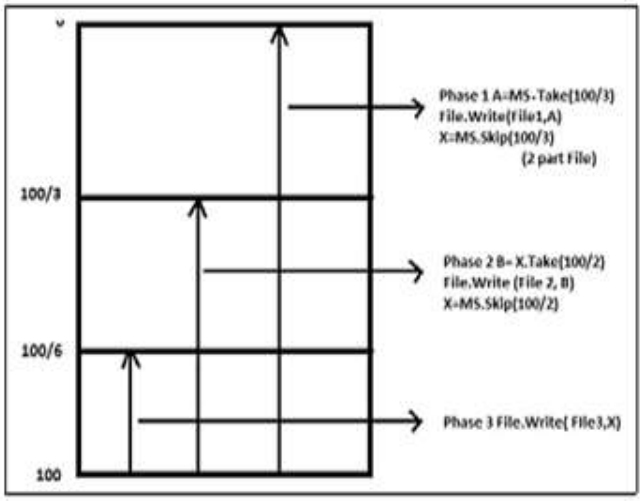

Figure5 .Splitting File

VIII. CONCLUSION

Once the system is ready it is expected that it should give the desired output of easy and efficient data access by the way of splitting and securing the data at multiple cloud environments without the knowledge of the end user. It should also expect that system should be capable to restore the data from remaining cloud storage in case of failure of any of the cloud storage. Another import ant expectation from the system is that it should be developer's friendly so that vast use of the platform by number of developers and it will results into multiple implementation of the syste*m.*

IX. REFERENCES

[1]"Lee Badger, Tim Grance, Robert Patt-Corner, Jeff Voas (May 2011) DRAFT Cloud Computing Synopsis and Recommendations", *NIST Special Publication 800-146*, Cloud Security Alliance (CSA) Released. December 17, 2009). Security Guidance for Critical Areas of Focus in Cloud Computing V2.1,http:// www.cloudsecurityalliance.org/guidance/ csaguide.v2.1.pdf

[2] "A Mordern Language for Mathematical Programming", Online at http://www.ampl.com.

[3] M. Arrington, "Gmail Disaster: Reports of mass emaildeletions",Onlineat http://www.techcrunch.com/2006/12/28/gmail-disasterreports-ofmass-email - deletions/, December 200

[4] P. S. Browne, "Data privacy and integrity: an overview", In Proceeding of SIGFIDET '71 Proceedings of the ACM SIGFIDET (now SIGMOD), 1971.

[5] A. Cavoukian, "Privacy in clouds", Identity in the Information Society, Dec2008.

[6] J. Du, W. Wei, X. Gu, T. Yu, "RunTest: assuring integrity of dataflow processing in cloud computing infrastructures", In Proceedings of the 5thACM Symposium on Information, Computer and Communications Security (ASIACCS '10), ACM, New York, NY, USA, 293-304.

[7] R. Gellman, "Privacy in the clouds: Risks to privacy and confidentiality from cloud computing", Prepared for the World Privacy Forum, online at http://www.worldprivacyforum.org/pdf/WPF Cloud Privacy Report.pdf,Feb 2009.

[8] The Official Google Blog, "A new approach to China: an update", online at http://googleblog.blogspot.com/2010/03/new-approach-to-chinaupdate.html, March 2010.

[9] S. H. Shin, K. Kobara, "Towards secure cloud storage", Demo for CloudCom2010, Dec 2010.

[10] C. Wang, Sherman S.-M. Chow, Q. Wang, K. Ren, W. Lou, "Privacypreserving public auditing for secure cloud storage", in InfoCom2010, IEEE, March 2010.