Improving Cloud Data Storage using Data Partitioning and Data Recovery using Seed Block Algorithm

Minaaz Shaikh, Aiswarya Achary, Sneha Menon, Nambirajan Konar

Department of Computer Engineering
SKN-Sinhgad Institute of Technology and Science,
Lonavala, Maharashtra, India

Abstract:- **Cloud storage provides online storage where data stored in form of virtualized pool that is usually hosted by third parties the partitioning method is proposed for the data storage which avoids local copy and reduces load on server. This method ensures high cloud storage integrity, security. To achieve this, remote data integrity checking concept is used to enhance the performance of cloud storage. To maintain this data efficiently, there is a necessity of data recovery services. We use a smart remote data backup algorithm, Seed Block Algorithm (SBA). The objective of proposed algorithm is to recover data ; first it help the users to collect information from any remote location in the absence of network connectivity and second to recover the files in case of the file deletion or if the cloud gets destroyed due to any reason.**

Keywords— Data Integrity Checking, Partitioning, Cloud Storage, Central Repository; Remote Repository; Secured hash function; Seed Block.

I. INTRODUCTION

In cloud computing, data is stored on the cloud server where we can access or retrieve information at any place and any time and the data is secured. Cloud service comprises of different types of services Software as a service (Saas), Platform as a Service(Paas) and Infrastructure as a service. There are various examples of cloud service like Amazon, IBMs blue cloud as IaaS; Google App Engine and Yahoo pig are Represented as PaaS .We cannot carry our laptop wherever we go hence cloud computing plays very vital role in storing our important files, retrieving when we needed at any place and anytime. Our data is secured from various user but it is not secured from the cloud employees that have credentials to access or retrieve our important files stored on cloud.

In the proposed system, Integrity is Analyzed when user uploads the file on storage and when it downloads. As shown in the Fig [1], client server works in cloud using web protocols.

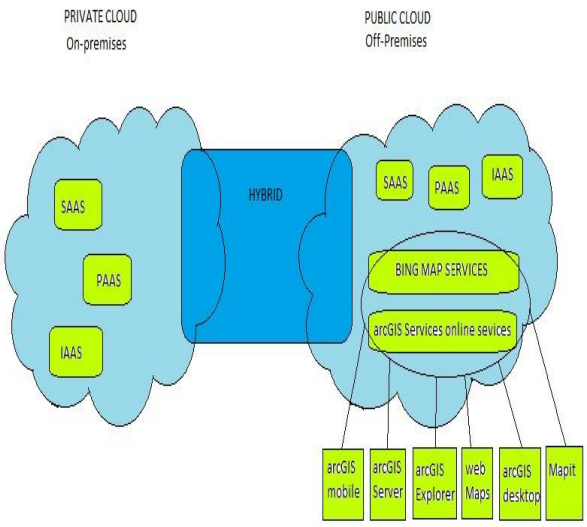


Fig.1. Cloud Services Architecture

II. PROBLEM STATEMENT

With cloud computing, organizations can use services and data is stored at any physical location outside their own control. This facility raised the various security questions like privacy, confidentiality, Integrity etc. and demanded a trusted computing environment wherein data confidentiality can be maintained. To induce trust in the computing,there is need of a system which performs authentication, verification and encrypted data transfer, hence maintaining data confidentiality [3].

Primary source of threats that are encountered are employees or insider attack and malicious hackers or outsider attack. Both can reach the private data and penetrate the security system. The overall aim of the proposed system is to secure the data stored on cloud, provide data integrity and recover data from storage server.

III. LITERATURE REVIEWS

In the Data Partitioning Technique literature review is done for data integrity checking and data storage mechanisms that are currently used in dynamic multi transactional applications. The dynamic data storage with token pre-computation and AES algorithm how it is stored in cloud is analysed. Integrity checking concepts is also used to detect avoid misbehaving server considering data correction and error localization. Distributed scheme is used to achieve the data quality, availability, integrity of dependable storage services. The data storage using dynamic data operation method is used to perform various operations. Security analysis is done by RSA to encode the data. Distributed storage system is also used to support the forwarded data in cloud without retrieval, ensuring secured and robust data in cloud storage.Data integrity in cloud storage devices are analyzed in the research works [8],[12]. Dynamic data operation and public Auditability are used for supporting the data integrity. The objective of this work is to have independent perspective and quality in services evaluating with the third party auditor. Storage model is also devised here to support multiple auditing tasks to improve efficiency. We consider generating signature methods for ensuring the cloud storage security. Dynamic operations are supported by using the RSA method. This method discusses data integrity and data correctness stored in cloud.

IV. EXISTING SYSTEM-PARTITIONING DATA

Data is stored in cloud computing and its security and Integrity is maintained using this proposed system. Data is stored, partitioned and then it is stored in servers.

A. Storage

In cloud storage, several users upload the file on the server with different file size. In the figure shown. The file is uploaded to an agent and then acknowledgment is given to user in return after the agent storing data on cloud. The proposed architecture ensures precomputation for data security,
Integrity.

In precomputation the security key is generated by then encryption technique to ensure security from unauthorized access. Public and private key is generated by encryption and decryption technique to ensure security.Data integrity functionis the important function in cloud storage.Normally when the data is stored the end users have to check whether the data is stored in cloud correctly or not. By the integrity checking process, the data is stored with security. Data are handled by remote data integrity checking; they do precomputation process to avoid threats [1].

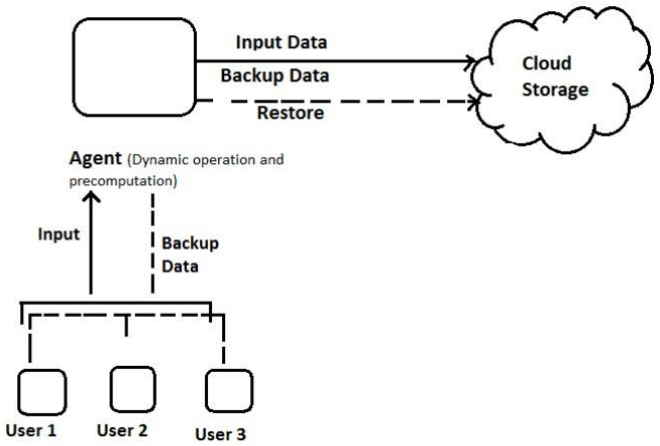


Fig 2. Cloud Storage

B. Accessing cloud data

When user wants to download the file from the server, the stored data is decrypted and then with the users private key data is retrieved. The user can share the data which it wants to share with others. Remote data integrity checking is used to maintain the data from threats. It also manages the effective storage and retrieval processes. The public auditability method manages the error localization,verification,misbehaving server and error recovery.This ensures data security from unauthorized access.It also increases the performance. Flexible access control is also provided for authentication in this work and to detect the attacks [1].

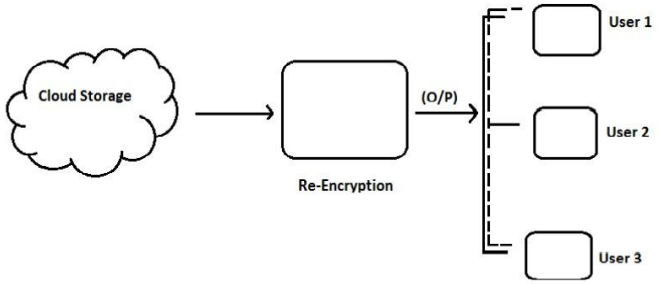


Fig 3. Accessing cloud

C. Partitioning

Large data files are partitioned and stored in cloud servers, partitioning helps in breaking down larger files into smaller modules and it reduces burden on storage server. Partitioned takes automatically when file is uploaded and then while retrieving the file it is merged and provided to user.

Algorithm for partitioning and merging files:-

1. File is uploaded to the cloud storage server.
2. Split files into n blocks with extension and size.
3. Encrypt the partitioned file

4. Merge the Encrypted partitioned file with its index value.
   Return file
5. Decrypt the merged file.

V. PROPOSED SYSTEM

In our proposed system the file is partitioned and it is send to the TPA server Fig [4]. The partitioned data is XORed and the data is stored in the Recovery server. The TPA server then creates digital signature using SHA1 algorithm, encryption and decryption of partitioned data is done.When the user retrieves file, the data is decrypted and the file is provided to user Fig [5].

As discussed above low implementation complexity, low cost, security and time related issues are still challenging in the field of cloud computing.To tackle these issues we propose SBA algorithm and in forthcoming section, we will discuss the design of proposed SBA in detail [2]. There are many techniques that have focused on these issues. In forthcoming section, we will be discussing a technique of back-up and recovery in cloud computing domain that will cover the foresaid issues.

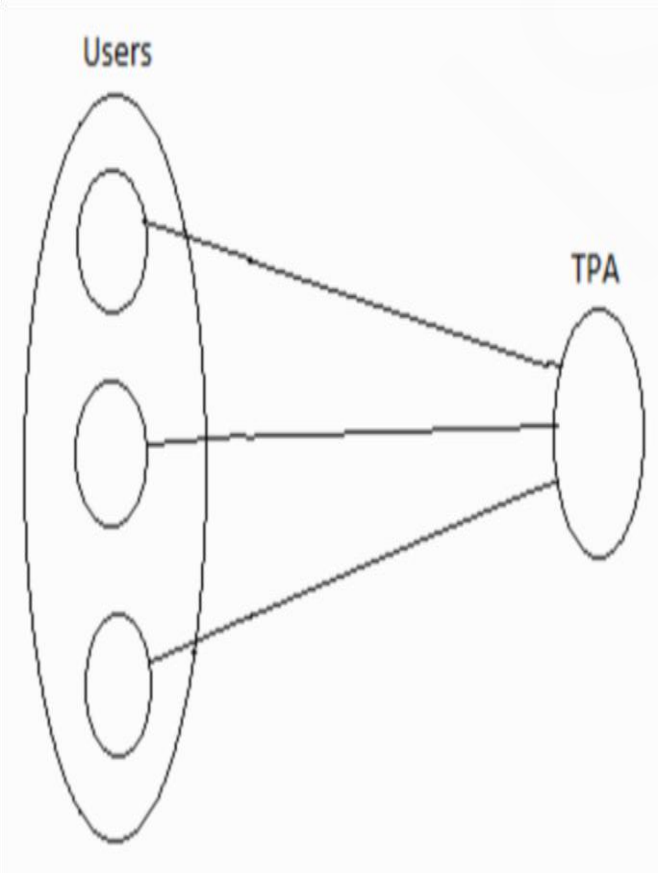


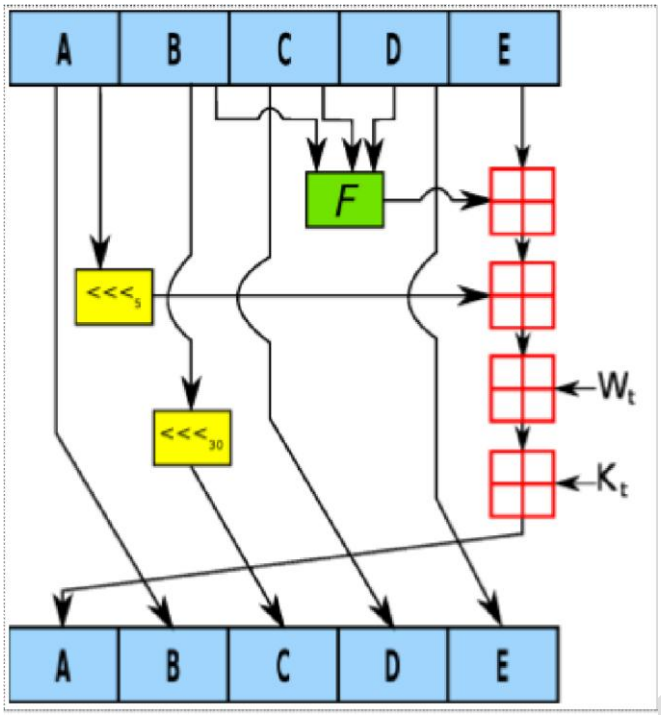
Fig 4. Uploading file on TPA server

A. Seed Block Algorithm

Seed block algorithm is used for back-up and recovery services. This algorithm simply uses EXOR functions that is when the data is partitioned say A and B, then XORing of A and B is done. If A is altered or the server is down, the partitioned data A can be easily retrieved by XORing the partitioned data B and the XORed data.
For Example, consider input data=5
Partition the data as 2 and 3
Partition data A=2; Its binary form is : 0010
Partition data B=3: Its binary form is : 0011
A $\oplus$ B=0001
Suppose if the Storage server 1 is down which consist of partitioned data A, then the XORed data and partition data B is XORed to get the data B as:
0011 $\oplus$ 0001=0010 i.e. Partitioned data A

B. Secured Hashing Algorithm (SHA1)

In cryptography, SHA-1 is a cryptographic hash function designed by the National Security Agency and published by the NIST as a U.S. Federal Information Processing Standard. SHA stands for "secure hash algorithm". The three SHA algorithms are structured differently and are distinguished as SHA-0, SHA-1, and SHA-2. SHA-1 is very similar to SHA-0, but corrects an error in the original SHA hash specification that led to significant weaknesses. The SHA-0 algorithm was not adopted by many applications. SHA-2 on the other hand significantly differs from the SHA-1 hash function. SHA-1 is the most widely used of the existing SHA hash functions, and is employed in several widely-used security applications and protocols.SHA-1 produces a 160-bit message digest based on principles similar to those used by Ronald L. Rivest of MIT in the design of the MD4 and MD5 message digest algorithms, but has a more conservative design. The original specification of the algorithm was published in 1993 as the Secure Hash Standard, FIPS PUB 180, by US government standards agency NIST (National Institute of Standards and Technology). This version is now often referred to as SHA-0. It was withdrawn by NSA shortly after publication and was superseded by the revised version, published in 1995 in FIPS PUB 180-1 and commonly referred to as SHA-1. SHA-1 differs from SHA-0 only by a single bitwise rotation in the message schedule of its compression function; this was done, according to NSA, to correct a flaw in the original algorithm which reduced its cryptographic security. However, NSA did not provide any further explanation or identify the flaw that was corrected. Weaknesses have subsequently been reported in both SHA and SHA-1. SHA-1 appears to provide greater resistance to attacks, supporting the NSA's assertion that the change increased the security



One iteration within the SHA-1 compression function:
A, B, C, D and E are 32-bit words of the state;
F is a nonlinear function that varies;
$\lll_n$ denotes a left bit rotation by *n* places;
n varies for each operation;
W_t is the expanded message word of round t;
K_t is the round constant of round t;
$\boxplus$ denotes addition modulo 2^{32}.

Fig 5. SHA1 Flow

To maintain data integrity we create digital signature of particular data. Digital signature is used uniquely identify the data. When the data is altered, SHA1 is used.

C. Advanced Encryption Standard (Aes) Algorithm

There are terms that are frequently used throughout this paper that need to be clarified.

Block:
AES is a block cipher.This means that the number of bytes that it encrypts is fixed.AES can currently encrypt blocks of 16 bytes at a time; no other block sizes are presently a part of the AES standard. If the bytes being encrypted are larger than the specified block then AES is executed concurrently. This also means that AES has to encrypt a minimum of 16 bytes. If the plain text is smaller than 16 bytes then it must be padded.
Simply said the block is a reference to the bytes that are processed by the algorithm.

State:
Defines the current condition (state) of the block.That is the block of bytes that are currently being worked on. The state starts off being equal to the block, however it changes as each
round of the algorithms executes

ENCRYPTION

1. Create a Cipher object and Key.
2. Create a Secret key using cipher object.
3. Initialize it with secret key
4. Encrypt the files
5. Get recipient's public key and Create Cipher and initialize it for encryption with recipient's public key.
6. Create Sealed Object to seal session key using asymmetric Cipher and Serialize Sealed Object.
7. Return the encrypted files and serialized Sealed Object to Recipient

DECRYPTION

1. Get encrypted message and serialized Sealed Object.
2. Re-serialize Sealed Object.
3. Create Cipher object, and initialize it for decryption and generate private key.
4. Unseal the key using the asymmetric Cipher.
5. Create Cipher object and Initialize it with the recovered session key for decryption. 6. Decrypt the files for access.

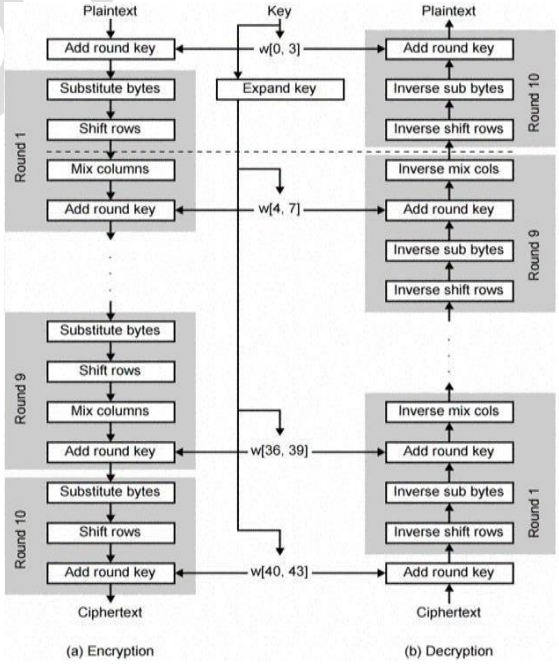


Fig 6. AES FLOW

After partitioning, encryption and creating digital signatures of the data the TPA servers then stores the data into storage servers as shown in the Fig [6].

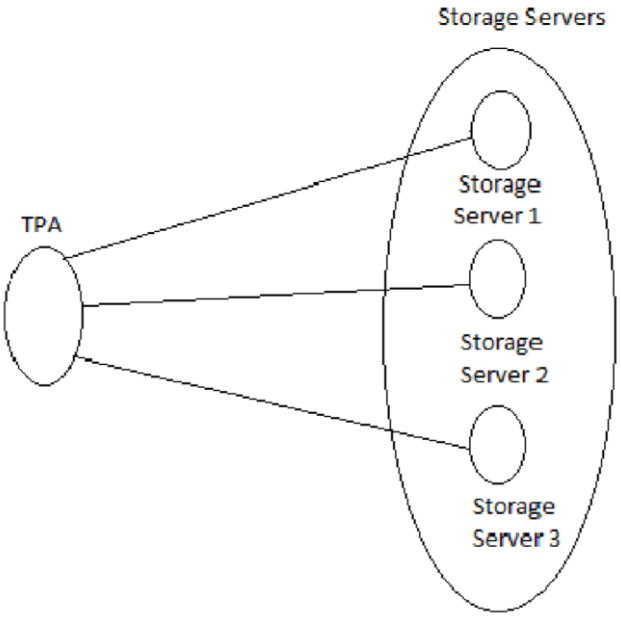


Fig 7. Storage to the Storage Servers

In the Proposed Architecture, Our system architecture consists of four storage servers, one client. There are two types of clients:
1) One who Uploads
2) One who Downloads

Client can register, login, download and upload file. Request for uploading and downloading is send to the TPA server i.e. the Third Party Application. TPA carries out the following functions: 1) Receive file
2) Partition file
3) Extract partition hash
4) Create backup partitions using Seed Block Algorithm
5) Encrypt partitions
6) Upload partition to individual server

TPA server is connected to three other servers i.e. the storage server1, storage server2 and the storage server3.Storage server1 consists of the first partitioned data and storage server2 consists of the second partitioned data of the original data. XORing of storage server1 and storage server2 is done and the resultant is stored in the storage server3.
In case any of the storage server is down then the recovery server is been XORed with the active server to respond to the data the client has requested. When the client uploads the file, the file gets partitioned and gets stored in the storage server1 and storage server2 also in the TPA the hashing of the data takes place where the data integrity is maintained. Now, the data is been stored into the storage servers. And whenever the data is to be downloaded then the TPA server downloads it From the storage servers and provides the required file to the client.

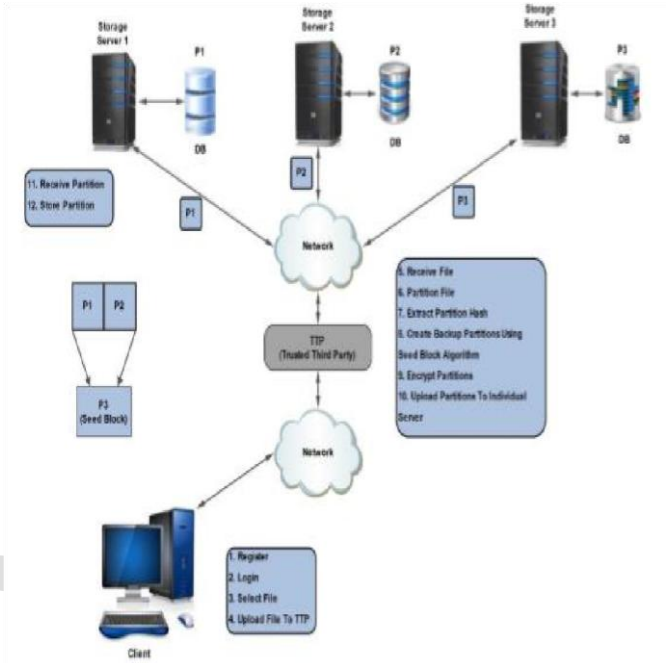


Fig 8. System Architecture

VI. CONCLUSION AND FUTURE WORK

We propose an efficient data storage security in cloud service. The partitioning of data enables storing of the data in easy and effective manner. It also gives way for flexible access and there is less cost in data storage. The space and time is also effectively reduced during storage. Dynamic operation is another key concept where, encoding and decoding process secures data, when storing into cloud. Also the remote data integrity checking detects the threats and misbehaving server while storing the data in cloud ensuring data security. There is no backup of the original data after partitioning. Hence there are high possibilities of data loss in case any of the servers goes down. Therefore to overcome the drawback we use the SBA algorithm. The proposed SBA is robust in helping the users to collect information from any remote location in the absence of network connectivity and also to recover the files in case of the file deletion or if the cloud gets destroyed due to any reason The time related issues are also being solved by proposed SBA such that it will take minimum time for the recovery process.

FUTURE ENHANCEMENT

It basically means an improvement that makes something agreeable in the near future. Future work is planned to

provide higher level of security and searching mechanisms for outsourced computations in cloud services. Here we are using only two storage servers, but in near future using HADOOP we can extend the number of storage servers as per our needs

REFERENCES

[1.] C. Selvakumar; G. Jeeva Rathanam; M. R. Sumalatha; "PDDS - Improving Cloud Data Storage Security Using Data Partitioning Technique" 2013 3rd IEEE International Advance Computing Conference (IACC), Dec 2013.

[2.] Ms. Kruti Sharma; Prof. Kavita R Singh;"Seed Block Algorithm: A Remote Smart Data Back-up Technique for Cloud Computing" 2013 International Conference on Communication Systems and Network Technologies, Dec 2013.

[3.] Zhiguo Wan; Jun'e Liu; Deng, R.H.; "HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing," Information Forensics and Security, IEEE Transactions on, vol.7, no.2, pp.743-754, April 2012.

[4.] Mr. Prashant Rewagad; Ms.Yogita Pawar; "Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing" 2013 International Conference on Communication Systems and Network Technologies, Dec 2013 11. William A. Arbaugh, Narendar Shankar and Y.C.Justin Wan, "Your 802.11 Wireless Network has No Clothes", University of Maryland, March 2001.

[5.] "Data Partitioning Technique to Improve Cloud Data Storage Security" ,.Swapnil V.Khedkar , A.D.Gawande Information Technology, Computer Science, SGBAU University Amravati, Maharashtra, India, Swapnil V.Khedkar (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3), 2014, 3347-3350