in the cost of storage hardware, the management of storage has become more complex and represents approximately 75% of the total ownership cost. Since the data owner physically releases sensitive data to a remote CSP [5], there are some concerns regarding confidentiality, integrity, and access control of the data. The confidentiality feature can be guaranteed by the owner via encrypting the data before outsourcing to remote servers [9]. The proposed model provides trusted computing environment by addressing important issues related to outsourcing the storage of data, namely confidentiality, integrity, access control and mutual trust between the data owner and the CSP. This means that the remotely stored. In summary, this paper shows that designed system provides inexpensive, Flexible solutions for shared network. They evaluate the system against Hadoop DB [2], an approach for data sharing applications which shows that the proposed system is significantly better than Hadoop DB.

2. Literature Survey

2.1. Existing System

A solution to detect cheating from owner side as well as CSP side is done through digital signatures. For each file owner attaches digital signature before outsourcing. The CSP first verifies digital signature of owner before storing data on cloud [7]. In case of failed verification, the CSP rejects to store data and asks the owner to resend the correct signature. If the signature is valid, both the file and signature are stored on the cloud servers. The digital signature achieves non-repudiation from the owner side. When an authorized user (the owner) requests to retrieve the data file, the CSP sends file, owner's signature and CSP's signature on (file || owner's signature). The authorized user first verifies the CSP's signature. In case of failed verification, the user asks CSP to re-perform the transmission process. If CSP's signature is valid, the user then verifies owner's signature. If verification fails, this indicates the corruption of data over the cloud servers [8]. The CSP cannot repudiate such corruption for the owner's signature is previously verified and stored by the CSP along with file. Since CSP's signature is attached with the received data, a dishonest owner cannot falsely accuse the CSP regarding data integrity. The above solution increases the storage overhead on cloud as owner's signature is stored along with the file on cloud servers. Moreover, there is an increased computation overhead, CSP has to verify signature of owner before storing file on cloud, and the authorized user verifies two signatures for each received file. If the CSP receives file from trusted entity other than the owner, the signature verification is not needed since the trusted entity has no incentive for repudiation or collusion [6]. Therefore, delegating small part of owner's work to the TTP reduces both the storage and computation overheads. However the outsourced data must be kept private and any leakage of data toward the TTP must be prevented Limitations are as follows,

a) The CSP is untrusted, and thus the confidentiality and integrity of data in the cloud may be at risk.
b) Computation Overhead is more in owner side As well as CSP side
c) A data owner and authorized users may collude and falsely accuse the CSP to get a certain amount of reimbursement.
d) The Owner May Loss the direct control over the sensitive data.

2.2. Cloud Computing and its Characteristics

Cloud Computing is identification of on-demand delivery of IT resources and programs through the Internet with pay-as-you-go manner. Cloud Computing makes efficient of accessing servers, storage, databases and also set of program services over the Internet. Cloud Computing providers like Amazon Web Services, Google Cloud Platform own and keep the network-connected hardware needed for these application services, while use and user provision what user need with a web application. Cloud computing can explain by its characteristics as follows:

a) On demand self service
b) Comprehensive network access
c) Resource pooling
d) Quick elasticity
e) Service that is measured

Service delivery in Cloud Computing contains three different service models, namely

a) Infrastructure-as-a-Service (IaaS)
b) Platform-as-a-Service (PaaS)
c) Software-as-a-Service (SaaS).

The three service versions or layer are finished by an end user layer that encapsulates the end user view on cloud services.

2.3. Peer to Peer Distributed Data Management

Practically all existing P2P systems are designed to support sharing of data at a coarse granularity (e.g., files, documents) [3]. It first distinguishes between P2P systems and distributed database systems. Then define P2P distributed data management by looking at three examples (due to space constraints) of how P2P technology can be employed for distributed database applications. This will also serve to motivate the need for database technology in P2P systems [6].

2.4. P2P Distributed Data Management Systems Applications

Peer to Peer Data Management System is explaining by giving some example of application field where these systems are used as follows [4].

a) Health Care: In a hospital, each specialist has a group of patients that are solely under his care, For most of patients, the specialist is willing to share their data, but there are always some cases that he is unwilling to share for different reasons (e.g., part of his research program on a new drug, etc). [5] By making the shareholder patient data available to other specialists, it allows them to look for other patients who may have similar symptoms as their own patients, and hence can help them in making better decisions on the

Licensed Under Creative Commons Attribution CC BY

treatment (e.g., drugs to prescribe, reactions to look out for, etc).

b) Genomic Data: The discovery of new proteins necessitates complex analysis in order to determine their functions and classifications [5][7]. The main technique that scientists use in determining this information has two phases. The first phase involves searching Known protein database. The second phase involves analysing the functions and classifications. While there are several known servers on genomic data (e.g., GenBank, SWISS-PROT and EMBL) [8], there are many more data that are produced each day in the many lab oratories all over the world. These scientists create their own local databases of their newly discovered proteins and results, and are willing to share their findings to the world! Clearly, this is an application for P2P distributed data management systems for the same reasons as the health care application.

c) Data Caching: In the above two examples, each participant is actively involved in the process of consuming and supplying data. P2P distributed data management can also be deployed in passive nodes: nodes that are used to share resources (storage or computational power) on data that they may or may not be interested. Caching results from earlier queries is one such example - a node may have issued a query to some server (e.g., a data warehouse), the results of the query can be cached on the node (or some other neighbouring nodes). In this way, another node that requests for data that overlap the query result can potentially obtain partial answers quickly from this node, and the remainder from the original server. This also lightens the load on the original server. Indeed, Kalnis et. al. [5] [8] [9] have shown how distributed caching can be deployed in P2P environments to speed up OLAP queries.

2.5. Query Processing

This system gives the two mode processing approach i.e parallel processing approach and adaptive processing approach [1]. Query is submitted over normal peer using fetching and processing. The normal peer remote the subquery and the results are shuffled to the query submitting peer P [11].

3. System Design

3.1 System Architecture

Figure 2 explains the system components and relationship between them.

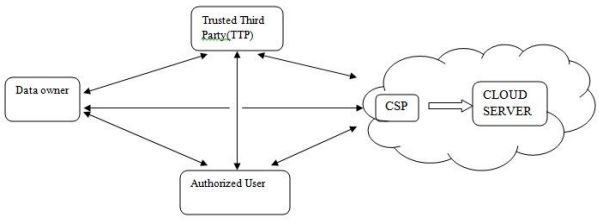

Figure 2: The System Architecture

3.1.1 Data Owner

Information Owner of the device component is the nothing the user of craving to save and share data over cloud. Information owner isn't having any idea where my information will be stored by the CSP and there is trust shortfall on CSP [10] [11]. As data is most important for info owner and the data owner do not desire that his information is observable to the CSP [12]. To fix the preceding issue set trustworthy third party and before uploading the data, it's encrypted / auditor which are set to keep watch.

3.1.2 Trusted Third Party / Auditor

Database auditing involves a database to not be unaware of the actions of the database users. Database administrators and consultants frequently set up auditing for the security purposes. For example to ensure that advice to be accessed by those without the permission do not access it. Auditing is the monitoring and recording of user database activities that are selected. It might be based on groupings of variables that can include user name, program, time, and so on, including the kind of SQL statement executed, or on individual activities [10] [12]. Auditing can be triggered by security policies when specified components including, within an Oracle database are obtained or altered the contents within a given object.

3.1.3. Authorized User

Authorized User is a client of owner who has right to access the remote data [12].

3.1.4. Cloud Storage Service Provider (CSP)

Database is provided by cloud Storage Services Provider. It permits information owner to keep any kind of information and also able to make the user define database schema. It can be Non SQL / SQL form of database instance. According to user requirement CSP will allocated the space for the user instance [12].

3.2 Module Design

3.2.1. The Data Owner Module

The Owner module can perform 4 operations. In upload operation, the data owner selects file F and generates a secret key k for a file. To achieve privacy-preserving, the owner creates an encrypted file $F' = E_k (F)$. The owner sends encrypted file to the TTP. TTP computes hash value for file $H(F')$ and sends file F' to CSP. Key response operation is used by the data owner to grant or revoke access to the outsourced file [8] [10]. In this operation, the data owner checks key requests from authorized users and if data owner wants to grant access then sends key.

3.2.2. Cloud Service Provider Module

The Cloud Service Provider (CSP) module is used to store and retrieve data. The CSP stores encrypted files F' sent by Owner and sends file to authorized users on demand [10].

Licensed Under Creative Commons Attribution CC BY

3.2.3. Trusted Third Party Module

The TTP module receives encrypted file F' from the data owner and computes hash value H(F') using SHA-1 algorithm [8]. It stores H(F') in its database which will be used during the dynamic operations and to determine the cheating party in the system (CSP or Owner). TTP send file F' to CSP module to store on cloud.

3.2.4. Authorized user module

Authorized users are set of owner's clients who have the right to access the remote data. To access the data, the authorized user sends a data-access request to the CSP and TTP, and receives the data file in an encrypted form F' from CSP and hash value of encrypted file H(F') from TTP [10]. To decrypt file authorized user requires secret key k generated by data owner. Authorized user sends key request to the data owner. The owner grants access to file by sending key k to user.

3.3 Algorithm

3.3.1. Algorithm for Creation of Bootstrap Peer

This algorithm is used for the building of this system using bootstrap peer node of having component of core use for building this system.

Algorithm 1 BootStrapDaemon ()
1: While true do
2: Status S = invokeCloudWatch ()
3: Array List PeerList = BootStrap.getAllPeer ()
4: Array List new Peer = New Array List ()
5: do i = 0 to PeerList. Size ()
6: if Peerlist.get (i).fails () then
7: Peer.loadMySQLBackUpFromRDS (PeerListget (i))
8: newPeer.add (peer)
9: BootStrap.setBlackList (PeerList.get (i))
10: else
11: if Peerlist.get (i).overloaded () then
12: Peer Peer = new Peer ()
13: BootStrap.removeAllNewPeer (Black List)
14: Sleep T second

3.3.2. Algorithm for Encryption

The encryption process consists of 10 rounds of processing for 128-bit keys For encryption, each round consists of these four measures: SubBytes (), ShiftRows (), MixColumns (), AddRoundKey ()

Algorithm 2 Encrypt ()
1: Cipher(byte[] input, byte[] output)
2: {
3: byte[4,4] State;
4: copy input[] into State[] AddRoundKey
5: for (round = 1; round < Nr-1; ++round)
6: {
7: SubBytes ShiftRows MixColumns AddRoundKey
8: }
9: SubBytes ShiftRows AddRoundKey
10: Copy State[] to output[]
11: }

4. Results and Analysis

4.1. Security Analysis

4.1.1 Detection of Dishonest Owner/User:

If the owner/user falsely faults the CSP regarding data integrity, the TTP performs cheating detection procedure. In this procedure, TTP retrieves encrypted file from CSP and computes the temporary hash value F1Htemp and compares F1HTTP and F1Htemp. If F1HTT =F1Htemp then F1 has not been corrupted on the server and owner/ user is dishonest.

4.1.2 Detection of Dishonest CSP:

During the data access phase of the proposed scheme, the authorized user receives the encrypted file F1 from the CSP and F1HTTP from the TTP. The authorized user computes hash of encrypted file F1Hu and associates F1HTTP and F1Hu.

If F1HTP /= F1Hu, a report is issued to TTP to determine the dishonest party. The TTP retrieves encrypted file from CSP and computes the temporary hash value F1Htemp and compares F1HTTP and F1Hu. If F1HTTP ≠F1Htemp, then F1 has been corrupted on the server and CSP is dishonest.

4.2. Performance Analysis:

The time performance of this paper was analysed under various file sizes. At first the time performance of this paper is evolved for different file sizes as shown in Table 1 and in Figure 3.

Table 1: Time Performance for file upload/download process

File Size	Upload(sec)	Download(sec)
1kb	3	2
10 kb	9	4
50 kb	10	6
150 kb	20	9
200 kb	25	13

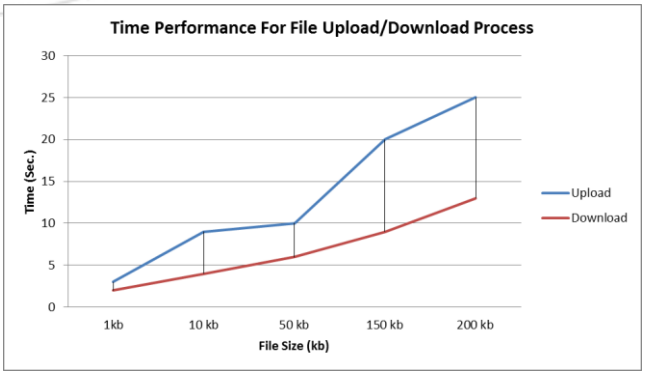

Figure 3: Time performance of Upload/Download Processes

5. Conclusion

This paper defines the important business model for storage of large scale date in shared network due to cloud computing

Licensed Under Creative Commons Attribution CC BY

storage system. Cloud computing storage system provide efficient way to access large scale data securely using encryption algorithm and provide credential security while access data in shared network. This system make portable using the combination of cloud computing storage system, database, and peer-to-peer technologies. This is created on Amazon EC2 cloud platform which can powerfully handle typical workloads in a shared network and can move near linear query throughput as the number of normal peers grows.

6. Future Scope

A number of future research directions stem from our current research. Problems to address during future research are Storage Overhead in TTP, In this work the files which are outsourced to the CSP from the data owner all these files has to store in the TTP, This is necessary in detection of Dishonest party, the storage space required to store the data is huge and it will take sustainable cost as well and also the maintenance of that particular data, The research may be proceeded to minimize the data stored in the TTP.

References

[1] Gang Chen, Tianlei Hu, Dawei Jiang, Peng Lu, Kian-Lee Tan, Hoang Tam Vo, and Sai Wu, "BestPeer++: A Peer-to-Peer Based Large-Scale Data Processing Platform", VOL. 26,NO. 6, JUNE 2014.
[2] H.V. Jagadish, B.C. Ooi, and Q.H. Vu, "BATON: A Balanced Tree Structure for Peer-to-Peer Networks," Proc. 31st Int'l Conf. Very Large Data Bases (VLDB '05), pp. 661-672, 2005.
[3] W.S. Ng, B.C. Ooi, K.-L. Tan, and A. Zhou, "PeerDB: A P2P-Based System for Distributed Data Sharing," Proc. 19th Int'l Conf. Data Eng., pp. 633-644, 2003.
[4] S. Wu, S. Jiang, B.C. Ooi, and K.-L. Tan, "Distributed Online Aggregation," Proc. VLDB Endowment, vol. 2, no. 1, pp. 443-454, 2009.
[5] S. Wu, J. Li, B.C. Ooi, and K.-L. Tan, "Just-in-Time Query Retrieval over Partially Indexed Data on Structured P2P Overlays," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '08), pp. 279-290, 2008.
[6] S. Wu, Q.H. Vu, J. Li, and K.-L. Tan, "Adaptive Multi Join Query Processing in PDBMS," Proc. IEEE Int'l Conf. Data Eng. (ICDE '09), pp. 1239-1242, 2009.
[7] Beng Chin Ooi, Yanfeng Shu, "Relational Data Sharing in Peer-based Data Management Systems." Kian-Lee Tan Sigmod Record special issue on P2P, 2003.
[8] B.C. Ooi, K.L. Tan, A.Y. Zhou, C.H. Goh, Y.G. Li, C.Y. Liau, B. Ling, W.S. Ng, Y.F. Shu, X.Y. Wang, M. Zhang " PeerDB: Peering into Personal Databases." The 2003 ACM SIGMOD Intl. Conf. on Management of Data (Demo). (SIGMOD 2003).
[9] Heng Tao Shen, Yanfeng Shu, and Bei Yu IEEE Trans. Knowl. "Efficient Semantic-Based Content Search in P2P Network." Data Eng. 16(7): 813-826(2004)
[10] K. E. Fu, "Group sharing and random access in cryptographic storage file systems," Master's thesis, MIT, Tech. Rep., 1999
[11] R. A. Popa, J. R. Lorch, D. Molnar, H. J. Wang, and L. Zhuang, "Enabling security in cloud storage SLAs with cloud proof," in Proceedings of the 2011 USENIX conference, 2011.
[12] Fu,"Plutus: Scalable secure file sharing on untrusted storage," in Proceedings of the FAST 03: File and Storage Technologies, 2003.

Author Profile

Miss. Bhavsar Harshada V completed her B.E. in Information Technology from Babasaheb Ambedkar Marat Wada University, Pune and doing M.E. from Sharadachandra Pawar College of Engineering, Dumberwadi.

Prof. G. D Deokate currently working as an assistant professor in SPCOE, Dumbarwadi.

Dr. S. V. Gumaste, currently working as Professor and Head, Department of Computer Engineering, SPCOE-Dumberwadi, Otur. Graduated from BLDE Association's College of Engineering, Bijapur, Karnataka University, Dharwar in 1992 and completed Post- graduation in CSE from SGBAU, Amravati in 2007. Completed Ph.D (CSE) in Engineering & Faculty at SGBAU, Amravati. Has around 22 years of Teaching Experience.

Licensed Under Creative Commons Attribution CC BY