ISSN: 2320-1363

Achieve the cloud data integrity Using Distributed Verification and Dynamic Support Schemes

A.Ramya*1, K.Srujana*2

M.Tech (CSE) Student, Department of CSE, PEC, Kandukur, Dist: Prakasam, AP, India

Associate professor, Department of CSE, PEC, Kandukur, Dist: Prakasam, AP, India

ABSTRACT:

Dynamic data support systems of cloud data storage distribute the data to users with high quality and correctness. Cloud data storage data maintains as security storage data. Previous system controls the internal and external threats. Some of the new problems are generated in transmission time. Some of data loss accidents are generated here. These problems are shows the different disadvantages like more incentives utilization. Cloud servers are not provides long terms services distribution. Present Cloud servers are not provides strong assurance for data integrity. Next introduces the cryptographic concepts provides the good security. It can contains some disadvantages, for overcome those problems introduces the TPA (third party auditor) concept. It is not gives the result as a performance. It is not handle the problems like server failures here. Sometimes data loss problems are generated here. It can give the unclear result.

The above all problems we are overcome in proposed system using cooperative data centers or distributed data centers mechanism. Access the data from any one of the server, server shows the errors. Those errors we are identifies at which block it is occur. Using dynamic operations implementation select another servers recover the error block of content. This is called as a distributed data collection. It can give the clear data distribution content. It is highly efficient compare to all previous techniques.

Index Terms: Assurances, distributed clouds, cryptographic techniques, data dynamics, cloud computing.

I.INTRODUCTION

Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers. Service providers create cloud computing systems to serve common business (or) research needs. Examples of cloud computing services include:

- *virtual IT* - configure and utilize remote, third-party servers as extensions to a company's local IT network
- *software* - utilize commercial software applications, or develop and remotely host custom built applications
- *network storage* - back up or archive data across the Internet to a provider without needing to know the physical location of storage.

Cloud computing systems all generally are designed for *scalability* to support large numbers of customers and surges in demand. Cloud storage is not just a third party data warehouse. The data stored in the cloud may not only be Recently, the importance of ensuring the remote data integrity has been highlighted by the following research works under different system and security models. These techniques, while can be useful to ensure the storage correctness without having users possessing local data, are all focusing on single server scenario. They may be useful for quality-of-service testing, but does not guarantee the data availability in case of server failures. Although direct applying these techniques to distributed storage (multiple servers) could be straightforward, the resulted storage verification overhead would be linear to the number of servers. As an complementary approach, researchers have also proposed distributed protocols for ensuring storage correctness across multiple servers or peers. Our work is among the first few ones in this field to consider distributed data storage security in cloud computing. Our contribution can be summarized as the following three aspects:

1. Compared to many of its predecessors, which only provide binary results about the storage status across the distributed servers, the proposed scheme achieves the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving server(s).

2. Unlike most prior works for ensuring remote data integrity, the new scheme further supports secure and efficient dynamic operations on data blocks, including: update,delete, and append.

3. The experiment results demonstrate the proposed scheme is highly efficient. Extensive security analysis shows our scheme is resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks. In existing system some internal and external threats. Some of the new problems are generated in transmission time. Some of data loss accidents are generated. These problems are shows the different disadvantages like more incentives utilization. Cloud servers are not provides long terms services distribution. Present Cloud servers are not provides strong assurance for data integrity. Next introduces the cryptographic concepts provides the good security. It can contains some disadvantages, for overcome those problems introduces the TPA (third party auditor) concept. It is not gives the result as a performance. It is not handle the problems like server failures here. Sometimes data loss problems are generated here. It can give the unclear result.

In proposed system, using cooperative data centers or distributed data centers mechanism. Access the data from any one of the server, server shows the errors. Those errors we are identifies at which block it is occur. Using dynamic operations implementation select another servers recover the error block of content. This is called as a distributed data collection.It can give the clear data

ISSN: 2320-1363

distribution content. It is highly efficient compare to all previous techniques.

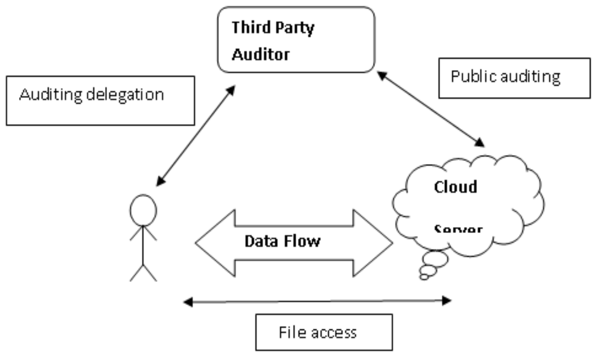


Fig. 1: Cloud storage service architecture.

II.RELATED WORK

Basically any web application/site that store "data" for you without you being aware of where/how exactly it's being stores is "cloud computing". The data goes inside a cloud & when required. comes out of that cloud. As a user you can't see (or don't care to) the inside working of thecloud.Forexample.,

1. Email servers like Yahoo & GMail store your mails which may also have file attachments too. Some enterprises rent such Email services for internal use from other vendors to keep their capital cost low. (Email servers & software are a bit costly to buy & maintain)

2. You can store files to servers on the web so that you can access them any where. How it is stored/retrieved is not important. e.g. DropBox

3. You want a database to store information but don't want the hassle to install one on your machine, you get that as well e.g. Microsoft Azure

4. You get CPU time too in cloud computing. So, if you were working for some complex problem & you temporarily require a server performing the computation for it, without requiring buy such high end servers, you get that too as part of cloud computing. e.g. Google Apps, Azure Apple has also introduced its own cloud computing which basically caters to email service & file storage service.

5. These techniques, while can be useful to ensure the storage correctness without having users possessing data, cannot address all the security threats in cloud data storage, since they are all focusing on single server scenario and most of them do not consider dynamic data operations. As an complementary approach, researchers have also proposed distributed protocols for ensuring storage correctness across multiple servers or peers. Again, none of these distributed schemes is aware of dynamic data operations. As a result, their

applicability in cloud data storage can be drastically limited.

III.PROBLEM STATEMENT

From the perspective of data security, which has always been an important aspect of quality of service, Cloud Computing inevitably poses new challenging security threats for number of reasons.

1. Firstly, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted due to the users' loss control of data under Cloud Computing. Therefore, verification of correct data storage in the cloud must be conducted without explicit knowledge of the whole data. Considering various kinds of data for each user stored in the cloud and the demand of long term continuous assurance of their data safety, the problem of verifying correctness of data storage in the cloud becomes even more challenging.

2. Secondly, Cloud Computing is not just a third party data warehouse. The data stored in the cloud may be frequently updated by the users, including insertion, deletion, modification, appending, reordering, etc. To ensure storage correctness under dynamic data update is hence of paramount importance.

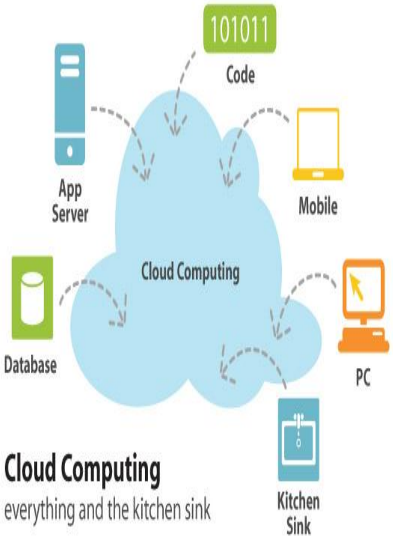
IV.PROPOSED SYSTEM ARCHITECTURE

In this paper, we propose an effective and flexible distributed scheme with explicit dynamic data support to ensure the correctness of users' data in the cloud. We rely on erasure correcting code in the file distribution preparation to provide redundancies and guarantee the data dependability. This construction drastically reduces the communication and storage overhead as compared to the traditional replication-based file distribution techniques. By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the storage correctness insurance as well as data error localization: whenever data corruption has been detected during the storage correctness verification, our scheme can

ISSN: 2320-1363

almost guarantee the simultaneous localization of data errors, i.e., the identification of the misbehaving server(s). There are five modules are there,They are:

1.System artwork:



In this module,Three different network entities are identified:

I.User: Who has data to be stored in the cloud and relies on the cloud for data storage.

II.CloudServer(CS): Which is managed by cloud service provider(CSP)to provide data storage service.

III.Third-Party Auditor(TPA): It is an optional,who has expertise and capabilities that users may not have,is trusted to assess and expose risk of cloud storage services on behalf of the users upon request.

2.File share ready:

It is well known that erasure-correcting code may be used to tolerate multiple failures indistributed storage systems, we rely on this technique to disperse the data file F redundantly across a set of n = m + k distributed servers. An (m, k) Reed-Solomon erasure-correcting code is used to create k redundancy parity vectors from m data vectors in such a way that the original m data vectors can be reconstructed from any m out of the m + k data and parity vectors. By placing each of the m + k vectors on a different server, the original data file can survive the failure of any k of the m + k servers without any data loss, with a space overhead of k + m. For support of efficient sequential I/O to the original file, our file layout is systematic, i.e., the unmodified m data file vectors together with k parity vectors is distributed across m + k different servers.

Algorithm 1:Token precomputation

```
procedure
Choose parameters l;n and function f, ϕ;
Choose the number t of tokens;
Choose the number r of indices per verification;
Generate master key $K_{PRP}$ and challenge key $K_{chal}$
For vector $G^{(J)}$,j ←1;ndo
for round i← 1;tdo
Derive $\alpha_i = f_{K_{chal}}$ (i) and $K_{PRP}(i)$ from $K_{PRP}$.
Compute $V_i^{(j)} = \varepsilon_{q=1}^{r} * G^{(j)}[\Phi_{K_{PRP}}^{(i)}$(q)]
end for
end for
Store all the vi's locally.
end procedure
```

V.Correct confirmation & failure locality:

Since our layout of file matrix is systematic, the user can reconstruct the original file by downloading the data vectors from the first m servers, assuming that they return the correct response values. Notice that our verification scheme is based on random spot-checking, so the storage correctness assurance is a probabilistic one. We can guarantee the successful file retrieval with high probability. On the other hand, whenever the data corruption is detected, the comparison of pre-computed tokens and received response values can guarantee the identification of misbehaving servers.

Algorithm 2: Correct confirmation & failure locality

```
procedure CHALLENGE(i)
Recompute $\alpha_i = f_{K_{chal}}$ (i) and $K_{PRP}(i)$ from $K_{PRP}$;
```

```
Send { $\alpha_i$ ,$K_{PRP}(i)$ }to all the cloud servers;
Receive from servers
{$R_i^{(j)}$=$\varepsilon_{q=1}^{r}\alpha_i{}^{q}$*$G^{(j)}$[$\phi_{K_{PRP}}^{(i)}$(q)]|1 $\leq j \leq n$}
for (j$\leftarrow m+1,n$)do
    $R^{(j)} \leftarrow R^{(j)}$-
    $\varepsilon_{q=1}^{r} f_{k_j}\left(s_{I_q},j\right).\alpha_i^{q}, I_q = \phi_{K_{PRP}}^{(i)}$(q
    )
end for
(( $R_i^{(1)},\ldots,R_i^{(m)},p == (R_i^{(m+1)},\ldots,R_i^{(n)}$))
 than
    Accept and ready for the next challenge.
else
    for (j ←1;n ) do
        if ($R_i^{(j)} != V_i^{(j)}$)than
             return server j is misbehaving.
        end if
    end for
end if
end procedure
```

VI.Document regain & failure regain:

User can reconstruct the original file by downloading the data vectors from the first m servers, assuming that they return the correct response values. Notice that our verification scheme is based on random spot-checking, so the storage correctness assurance is a probabilistic one. However, by choosing system parameters appropriately and conducting enough times of verification, we can guarantee the successful file retrieval with high probability. On the other hand, whenever the data corruption is detected, the comparison of precomputed tokens and received response values can guarantee the identification of misbehaving server(s) (again with high probability), which will be discussed shortly.

Algorithm 3:Error recovery

```
procedure
% Assume the block corruptions have been detected among
% the specified r rows;
% Assume $s \leq k$ servers have been identified misbehaving
Download r rows of blocks from servers;
Treat s servers as erasures and recover the blocks.
Resend the recovered blocks to corresponding servers.
end procedure
```

VII.Third Party Auditing

As discussed in our architecture, in case the user does not have the time, feasibility or resources to perform the storage correctness verification, he can optionally delegate this task to an independent third party auditor, making the cloud storage publicly verifiable. However, as pointed out by the recent work, to securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy. Namely, TPA should not learn user's data content through the delegated data auditing.

Cloud operations:

Update Operation

In cloud data storage, sometimes the user may need to modify some data block(s) stored in the cloud, we refer this operation as data update. In other words, for all the unused tokens, the user needs to exclude every occurrence of the old data block and replace it with the new one.

Delete Operation

Sometimes, after being stored in the cloud, certain data blocks may need to be deleted. The delete operation we are considering is a general one, in which user replaces the data block with zero or some special reserved data symbol. From this point of view, the delete operation is actually a special case of the data update operation, where the original data blocks can be replaced with zeros or some predetermined special blocks.

Append Operation

In some cases, the user may want to increase the size of his stored data by adding

ISSN: 2320-1363

blocks at the end of the data file, which we refer as data append. We anticipate that the most frequent append operation in cloud data storage is bulk append, in which the user needs to upload a large number of blocks (not a single block) at one time.

VIII. CONCLUSION

Cloud computing is still struggling in its infancy with negative and positive Comments. Academia is bit slower to react. Its security deficiencies and benefits need to be carefully weighed before making a decision to implement it.

IX.REFERENCES

[1] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th Int'l Workshop Quality of Service (IWQoS '09), pp. 1-9, July 2013.

[2] Amazon.com, "Amazon Web Services (AWS)," http://aws.amazon.com, 2009.

[3] Sun Microsystems, Inc., "Building Customer Trust in Cloud Computing with Transparent Security," https://www.sun.com/ offers/details/sun_transparency.xml, Nov. 2009.

[4] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69-73,2012.

[5] M. Arrington, "Gmail Disaster: Reports of Mass Email Deletions,"http://www.techcrunch.com/2006/12/28/gmail-disasterreportsof-mass-email-deletions, Dec. 2006.

[6] J. Kincaid, "MediaMax/TheLinkup Closes Its Doors," http://www.techcrunch.com/2008/07/10/mediamaxthelinkup-closesits-doors, July 2008.

[7] Amazon.com, "Amazon S3 Availability Event: July 20, 2008,"http://status.aws.amazon.com/s3-20080720.html, July 2008.

[8] S. Wilson, "Appengine Outage," http://www.cio-weblog.com/50226711/appengine_outage.php, June 2008.

[9] B. Krebs, "Payment Processor Breach May Be Largest Ever,"

http://voices.washingtonpost.com/securityfix/2009/01/payment_processor_breach_may_b.html, Jan. 2009.

[10] A. Juels and B.S. Kaliski Jr., "PORs: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security(CCS '07), pp. 584-597, Oct. 2007.

[11] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z.Peterson, and D. Song, "Provable Data Possession at UntrustedStores," Proc. 14th ACM Conf.

Computer and Comm. Security (CCS'07), pp. 598-609, Oct. 2007.

[12] M.A. Shah, M. Baker, J.C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest," Proc. 11th USENIX Workshop Hot Topics in Operating Systems (HotOS '07), pp. 1-6, 2007.

[13] M.A. Shah, R. Swaminathan, and M. Baker, "Privacy-Preserving Audit and Extraction of Digital Contents," Cryptology ePrint Archive, Report 2008/186, http://eprint.iacr.org, 2008.

[14] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. Fourth Int'l Conf. Security and Privacy in Comm. Netowrks (SecureComm '08), pp. 1-10,2008.

[15] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," Proc. 14th European Conf. Research in Computer Security (ESORICS '09), pp. 355-370, 2009.