Volume: 5 Issue: IV Month of publication: April 2017

DOI: http://doi.org/10.22214/ijraset.2017.4169

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Multi copy dynamic data in cloud Computing

Dr. Gowaramma Y. P[1], Omprakash B[2]
[1]Professor, Department Of CSE, Department Of CSE, KIT, Tiptur
[2]Assistant Professor, Department Of ISE, Atria IT, Bengaluru

Abstract: cloud computing is used over network of host servers on Internet to store, process, share and manage rather than local server or personal computer. And there is rapid increase of organization opting for outsourcing data to remote cloud service provider. Customers should pay for the cps storage infrastructure to store and retrieve unlimited amount of data by paying in the measures of gigabytes/month. For more scalability, availability and durability.

User might even want their data to get replicated on multiple server across different data centers. More of the copies, more the customer is asked to pay. So, the customers need to have a strong belief that the cps will store all the data copies that are agreed upon in the contract and all the copies should be synchronized with the modifications that has been done by the customers. In this paper, we are trying to propose a map-based provable multi copy dynamic data possession (MB-PMDDP) scheme that has the respective features:

The customers will have an proof that the services provider is not cheating by storing only fewer copies

The customers will be supported with block-level operation,i.e. block-level modification ,insertion, deletion and append.

Only authorized users can access the file copies stored by the cps. The proposed MB-PMDDP is an extension of dynamic single copy schemes. This analysis will be validated on experimental results on a commercial cloud platform and also provide security against clouding serves and identify the corrupted copies by addition modification in the proposed scheme.

I. INTRODUCTION

Cloud computing provides various computing services like shared computer processing resources and data to computers and other devices on demand. Cloud provides benefits like cost, speed, global scale, productivity, performance and reliability.

Since most of the organization are dependent on cloud ,it is important to ensure the loss or corruption of data. Many works [1] [2] [3] [4] [5] [6] [7] [8] have been done on designing remote data integrity checking protocol. By this we can check the integrity without downloading the complete data. [9] After going through lot of works and papers they have proposed a remote storage auditing method based on precomputed challenge response pairs. At present ,many works [10] [11] focuses on providing three features for remote data integrity checking protocols:

A. Data integrity [5] [7].
B. Public verifiability [2] [7].
C. Privacy against verifiers [8] [11].

Not only providing integrity we should also provide authenticity. Checking the authenticity of data has a created issue in storing data on untrusted servers. It usually—occurs in peer-to-peer storage systems [12] [13] ,network file system [14] [15],long term archives [16],web service object stores[17] and database system [18].These systems prevent storage servers from misrepresenting or modification by providing authenticity verification when accessing data.

The archival storage should be provided with guarantees about the authenticity of data on storage,like if storage servers possess data. The archival storage is insufficient to detect or identify if the data has been modified or deleted while accessing the data,because it will be too late recover lost or damaged data. However this PDP should provide the clients that should be able to verify that a server has retained file data without retrieving the data from the server and without having the server access the entire file. The previous techniques do not provide solutions for PDP.

Some schemes [19] provide a weaker guarantee by enforcing storage complexity. So,we define a model for PDP that provides proof that a third party stores a file. This model is so unique that it allows the server to access small portions of the file in generating the proof than accessing the entire file.

PDP only concentrates on static single files so we are trying to view the problem of creating multiple unique replicas of a file in multiple storage system. The main intention is to give data owner an archive data that is present in third party storage like amazon [20] or storage request broker [21] to introspection and maintain on its data. This MRPDP can be applied to all replication based,distributed and unstructured storage system,that includes peer-to-peer storage systems [22] [23] [14] [25] [13].

Replication also ensures the availability and durability of data [27]. Organizing the number and placing the replicas is critical to this process. When the system re-replicate data,replicas fail [28] [29],evaluates the correctness of replicas in the system[30] and moves replica among sites to meet availability [31] [32].

We have two dynamic multi-copy provable data possessions schemes that the end user can have an guarantee that the csp will store all the copies that are agreed upon during the service contract. It also concentrates on outsourcing dynamic data that would be block modification,insertion,deletion and append. The user can access the copies any number of time.

We even prove the correctness of our scheme against clouding servers. Cloud servers can provide response to the users challenges if and only if they actually have all data copies in an uncorrupted manner and updated state.

We also present theoretical analysis and experimental results to justify the performance of the proposed schemes.

II. ISSUES

A. Focus on a single copy of the data
B. Provide no guarantee that the CSP stores multiple copies of customers’ data
C. Provide no evidence that the copies stored are not modified in any way

III. IDENTITY BASED PROVABLE DATA POSSESSION IN MULTI-CLOUD STORAGE

Remote data integrity is used in cloud computing. It can make the clients check whether their sent data is kept intact without downloading the whole data. We are using a novel remote data integrity checking model i.e. ID-DPDP: Identity-based Distributed Provable Data Possession in multi-cloud storage. This ID-DPDP protocol may be secured under the assumption of the standard CDH (computational Diffie- Hellman) problem. On the client’s authorization the ID-DPDP protocol can identify

A. Private verification
B. Delegated verification
C. Public verification.

The issue to satisfy the cloud clients that their data are kept intact is important since the clients do not store these data locally. Remote data integrity checking is a primary issue. When the customer stores his data on multi-cloud servers, the distributed storage and integrity checking are absolutely necessary. The integrity checking protocol must be efficient in order to make it compatible for capacity-limited end user devices. Thus, based on distributed computation, we will go through distributed remote data integrity checking model and the corresponding concrete protocol will be presented in multi-cloud storage.

D. System Model of ID-DPDP.

An ID-DPDP protocol comprises four different entities as shown in Figure 2.1:

1) *Client:* An entity, which has large amount of data to be store on the multi-cloud for maintaining and manipulation, can be either single consumer or an organisation.
2) *CS (Cloud Server):* An entity, which is mantained by cloud service provider, has storage space and manipulation resource to maintain the consumer’s data.
3) *Combiner:* An entity, which receives the storage request and assigns the block-tag pairs to the respective cloud servers. When receiving the request, it splits the request and assign them to the different cloud servers. While
4) receiving the responses from the cloud servers, it combines them and delivers the combined response to the verifier.PKG (Private Key Generator): An entity, that receives the identity and it outputs with its respective private key.

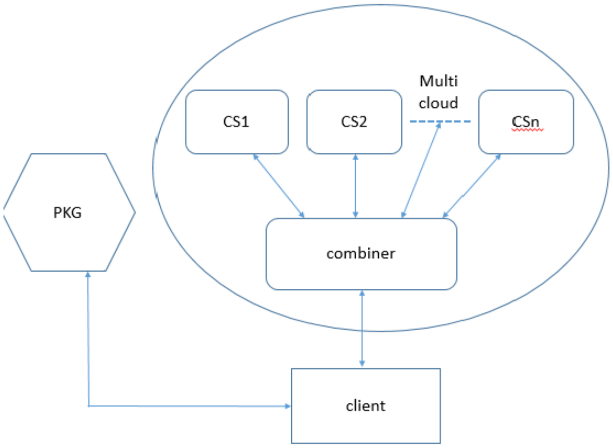


Fig 2.1 System Model of ID-DPDP

E. An Efficient & Secure Protocol For Data Storage In Cloud

There are few existing remote integrity checking methods which serves static data and cannot be applied to the report service since the data in the cloud can be dynamically updated. Thus, an effective and secure dynamic auditing protocol is required to satisfied data owners that the data is correctly stored in the cloud. We primarily design a reporting framework for cloud storage systems and introduce an efficient and privacy-preserving auditing protocol. The auditing protocol is extended to imply the data dynamic operations. Which are efficient and secure in the random model. The analysed and simulated report show that the implemented auditing protocols are secured and efficient. This project will propose a good scheme to reach optimised safety of data from the third party with the use of cryptographic method. In security issue it has developed on the significance of ensuring remote data integrity. Various security problems like, data loss that occurs in cloud computing. The cryptographic measures cannot be implied directly. The verification of data that is done free from the actual data which is a huge drawback.

The data stored in cloud is open for the attackers no matter how much the data is secured. The data is been isolated from the encryption and decryption processes from the cloud to a service that is been trusted by both the cloud provider and the cloud consumer for highly secured and protected data. To obtain maximum security, the data has been divided and encrypted with the help of highly secured processors .so, that the data is protected from attackers. It uses a protocol implying Sobol sequence and ECC for integrity and security of the data available in the cloud which are far impressive than those of RSA and other PKC methods. This design and analysis also proposes a scheme of modify, insert or delete, organise the data, stored in the cloud. In the design, the encryption of the data is done to ensure the integrity and then, the computation of metadata is done over the encrypted data. This is accomplished only when the user demands it.

Cloud Storage Model: User is the person who uses the services of cloud.

Cloud Service Provider (CSP): The data to be stored or retrieved through CSP. CSP manages the cloud server and provides a paid service to the customer.

Third Party Auditor (TPA): TPA is also called a Verifier; if the user is suffering from lack of time then the data verification is done by TPA or verifier. The following concept explains the cloud storage model and working of each type. In cloud computing the user is the one who stores his private data in cloud to protect it from hazards and this is done with the help of CSP. If the user wants the information again in order to access a particular data, then the user have to deliver an appeal to the CSP .Later, the CSP will check whether the user is authorized or not. If the user is authorized then it will allow the user to access the data otherwise not. If the data accessed by the user is in encrypted form then it can be decrypted using his secret key. And the last the TPA will perform a periodic check on the data and verify it periodically only when the user himself allows to verify the TPA. The data may be lost or modified by unfair means. Thus to protect it from these many obstacles an efficient and secure process is needed.

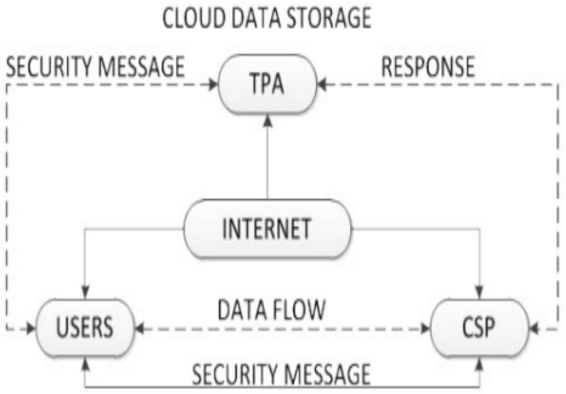


Figure 2.2 Cloud Storage Model

F.   Provable Possession and Replication of Data over Cloud Server Author

Outsourcing data to a remote Cloud Service Provider (CSP) is a growing trend for numerous customers and organizations increasing the burden of local data storage and maintenance. Moreover, users depend on the data replication provided by the CSP to guarantee the availability and life span of their data. Therefore, Cloud Service Providers (CSPs) allow storage infrastructure and web services interface that can be used to retrieve and store a large amount of data with fees metered in GB/month. The mechanisms used for data replication differ according to the nature of the data; more copies are needed for critical data that can't easily be reproduced. This critical data should be duplicated on multiple servers across multiple data centers. Also, non-critical, reproducible data are stored at lower levels of redundancy. Therefore, more importance is given to the customers to have a good evidence that they actually get the service they pay for. Moreover, they need to verify that all their data copies are not being corrupted with or deleted over time. The problem of Provable Data Possession (PDP) has been considered in all research papers. Unfortunately, the other PDP schemes focus on a single copy of the data and provide no guarantee that the CSP stores multiple copies of end users data. The correctness and completeness of end users data in the cloud is put at risk due to the following issues. Primarily,CSP —The main aim is to make profit and maintain the reputation —that has incentive to minimize data loss or reduce storage by removing data that has not been frequently accessed. Secondly, a greedy CSP can delete some of the data or might not store all data in fast storage required by the contract with certain users, That is, place it on Compact disks or other offline media and thus using less storage. Finally, the cloud infrastructures has a wide range of internal and external security threats. Examples of security attacks or breaches of cloud services appear constantly. In short, although outsourcing data into the cloud is economically attractive and the complexity of large-scale data storage, it does not provide any guarantee on data perfection. This problem, if not properly maintained, may have dip in the successful deployment of cloud architecture. Since customers' data has been outsourced to remote servers, efficient verification of the completeness and correctness of the outsourced data becomes a formidable challenge for data security in CC. We use cryptographic measures for data verification and availability based on hashing protocol and signature scheme are not applicable on the outsourced data without having a copy of the data. It is not possible for the clients to download all stored data in order to validate its integrity therefore clients need effective techniques to check the integrity of their outsourced data with low manipulation, communication, and storage read and write overhead. Therefore, many researchers have concentrated on the problem of Provable Data Possession (PDP) and proposed various schemes to check or verify the data stored on remote servers. Basically, Provable Data possession (PDP) is a measure for validating data integrity over remote servers. In this project we try to address this challenging problem and introduce two Efficient Multi-Copy Provable Data Possession (EMC-PDP) protocols, and prove the security (correctness) of our protocols against colluding servers. Provable data possession (PDP) is a technique for checking the integrity of data/information in outsourcing storage service. The fundamental aim of the PDP scheme is to allow the user to efficiently, periodically, and securely validate a remote server — which stores the owner's large amount of data — that is not cheating the verifier. The problem of data integrity over remote servers has been dealing for many years and there is a simple solution to solve this problem as follows. Primarily, the data owner computes an acknowledgement authentication code (MAC) of the file before outsourcing to a remote server. Then, the end user keeps only the computed MAC on his local storage, sends the file to the remote server, and deletes the local copy of the file. Later, whenever a user needs to check the data integrity, the user sends a request to retrieve the data from the archive service provider, re-manipulates the MAC of the file, and compares the re-computed MAC with the previous value. Another measure is, Instead of computing and storing the MAC of the file, the end users divides the file F into blocks {b1, b2, b3 . . bm}, computes a MAC σi for each block bi: $\sigma i = MACsk(i||bi)1\le i\le m$, sends both the data file F and the MACs $\{\sigma i\}1\le i\le m$ to the remote/cloud server, destroys the local copy of the file, and stores only the secret key sk. When the verification process is being processing, the verifier requests for a set of randomly selected blocks and their respective MACs, re-manipulates the MAC of each received block using sk, and compares the re-computed MACs with the retrieved values from the remote server. Behind the second approach is that checking part of the file is much easier than the entire file. However both approaches suffers from severe drawback; the communication complexity is linear with the data size which is not practical, especially when the bandwidth is limited. The data stored in cloud is open to the attackers no matter how much the data is secured and protected. It implies the isolation of the encryption and decryption processes from the cloud to a broker service that is trusted by both the cloud provider and the cloud consumer for maximised secured data. To attain maximum security, the data has been distributed and encrypted with the help of highly secured processors so that the data is protected from unfair means. It has implied a protocol using Sobol sequence and ECC for the integrity and the security of the data that is available in the cloud which is far better than those of RSA and PKC methods. One of the core design principles of outsourcing data is to provide dynamic behaviour of data for different applications. This means that the remotely stored data can be not only accessed by the users, but also update and scale the data (through block level operations) by the owner. PDP schemes presented earlier focus on only static or warehoused data, where the outsourced data is kept unchanged over remote servers. Examples of PDP constructions that uses the dynamic data also exist. But only for a single copy of the data file. Although PDP schemes have been projected for multiple copies of static data, but to the best of our survey, this work is the first PDP scheme that directly deals with multiple copies of dynamic data. When checking multiple data copies, the overall integrity verification fails if there is one or more corrupted copies. To solve this issue and identify which copies have been corrupted, this project discusses a differential modification to be applied to the proposed scheme.

IV. CONCLUSION AND FUTURE ENHANCEMENTS

To ease the burden of local data storing and maintenance, many organizations have opted the outsourcing data to remote servers. In the work, the problem of creating multiple copies of dynamic data files and verifying those copies stored on untrusted cloud servers are analyzed. We have proposed a new PDP scheme referred to as MB-PMDPP [31] which supports outsourcing of multiple copy dynamic data, where the data owners are able to update and scale these copies on the remote servers along with achieving and accessing the data copies stored by CSP. It is also important to reduce the computation time this can be done by using MB-PMDDP [31]. We conclude by saying that is important for any CPS provider to provide fast accessibility, availability, integrity and multiple copy of the data . And all these can be achieved by using the model MB-PMDDP [31]. In none of the previous paper they have talked about recovering corrupted files. A slight modification can be done on the proposed scheme to support the feature of identifying the induces of corrupted copies.

The corrupted data copy can be reconstructed even from a complete damage using duplicated copies on other servers. Through security analysis

REFERENCES

[1] f. sebe , j.domingo-ferrer, a.martinez-balleste, y.deswarte and j.-j .quisquater " effecient remote data possession checking in critical information infrastructure " ,aug 200

[2] g. ateniese ,r.burns, r.curtmola, j.herring,a.kissner ,z.peterson and d. song, " provable data possession at untrusted stores" 2007

[3] r.curtmula,o.khan,r.burns and g.ateniese "mr-pdp :multiple replica provable data possession" 2008

[4] g.ateniese ,r.di pietro,l.v.mancini and g.tsudik "scalable and efficient provable data possession" 2008

[5] c.erway, a.kupcu,c.papamanthou and r.tamassia "dyanmic provable data possession" 2009

[6] c.wang ,q.wang ,k.ren and w.lou "ensuring data storage security in cloud computing" july 2009

[7] q.wang,c.wang,j.ki ,k.ren and w.lou "enabling public verifiability and data dynamics for storage security in cloud computing" sept 2009

[8] c.wang ,q.wang ,k.ren and w.lou "privarcy -preserving public auting for data storage security in cloud computing " march 2010

[9] m.a.shah,m.baker,j.c.mogul and r.swaminathan " auditing to keep online storage services honest" 2007

[10] c.wang,s.s-m chow,q.wang,k.ren and w.lou "privarcy -preserving public auditing for secure cloud storage"2009

[11] .zhu h.wang,z.hu,g-j.ahn,h.hu and s.s.yau"comparative provable data possession"201

[12] .kubiatourcz,d.bindel,y.chen,p.eaton,d.cruls,r.crummadi,s.rhea,h.weahther spoon,w.weimer,c.wells and b.zhao "an architecture for gobal-scale persistent storage" nov 200

[13] a.a.muthita choroen,r.morris,t.m.cril and b.chen"read or write peer to peer file system"

[14] j.li,m.krohan,d.mazieres, d.shasha "secure untrusted data repository "(sundr) 2004.

[15] m.kallahalla,e.riedel,r.swaminathan,q.wang and k.fu.plutus "scalable secure file sharing on untrusted storage" 2003.

[16] p.maniatis,m.roussopoulos,t.giuli,d.rosenthal,m.baker and y.muliadi "the lockss peer to peer digital preservation system" 2005

[17] a.y.yumerefendi and j.chase strong "accountability for network storage",2007

[18] p.maheswari,r.vigralek, and w.shapiro " how to build a trusted database system on untrusted storage"

[19] p.crolle,s.jarecki and i.mironov "cryptographic primitives enforcing communication and storage complexity " 2002

[20] amazon simple storage services(amazon s3) . aws.amazon.com/s3

[21] srb-storage resource broker http://www.sdsc.edu/srb/index.php/main-page

[22] A. Rowstron and P. Druschel. Storage Management and "Caching in PAST, A Large-scale, Persistent Peer-to-peer Storage Utility". In Proc. of SOSP '01, 2001

[23] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Sto-ica. "Wide-area cooperative storage with CFS". In Proc. Of SOSP '01, 2001.

[24] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard. "A cooperative internet backup scheme". In Proc. of USENIX Technical Conference, 2003.

[25] A. Haeberlen, A. Mislove, and P. Druschel. Glacier: "Highly durable, decentralized storage despite massive correlated failures". In Proc. of NSDI '05, 2005.

[26] B.-G. Chun, F. Dabek, A. Haeberlen, E. Sit, H. Weather-spoon, M. F. Kaashoek, J. Kubiatowicz, and R. Morris. "Effi-cient replica maintenance for distributed storage systems". In Proc. of NSDI '06, 2006.

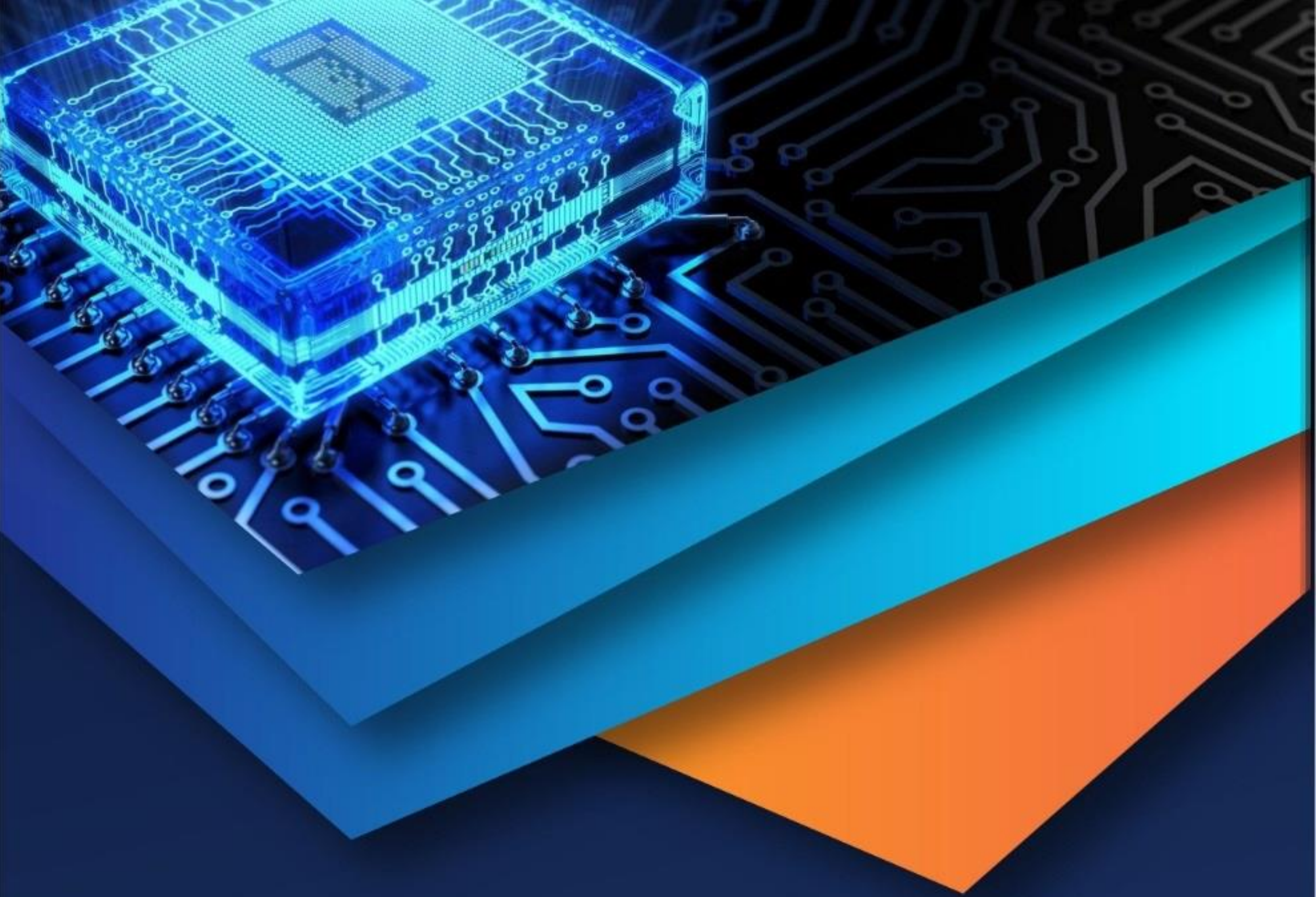
[27] F. Dabek, J. Li, E. Sit, J. Robertson, M. F. Kaashoek, and R. Morris." Designing a DHT for low latency and high throughput". In Proc. of NSDI '04, 2004.

[28] P. Maniatis, M. Roussopoulos, T. Giuli, D. Rosenthal,M. Baker, and Y. Muliadi. "The LOCKSS peer-to-peer dig-ital preservation system. ACM Transactions on Computing Systems", 23(1):2–50, 2005.

[29] A. Adya, W. Bolosky, M. Castro, R. Chaiken, G. Cermak,J. Douceur, J. Howell, J. Lorch, M. Theimer, and R. Watten-hofer. Farsite: "Federated, available, and reliable storage for an incompletely trusted environment". In Proc. of OSDI '03,2003.

[30] W. J. Bolosky, J. R. Douceur, D. Ely, and M. Theimer. "Feasability of a serverless distributed file system deployed on an existing set of desktop PCs". In Proc. of ACM SIG METRICS, 2000.

[31] MF-PDP: Multi-function provable data possession scheme in cloud computing Xiaojun Yu; Qiaoyan Wen,2014



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429