ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 10, December 2012

A Comparative Survey on Availability and Integrity Verification in Multi-Cloud

Ms.V.Mangaiyarkkarasi† and Mr.K.A.Dhamodaran††

†M.E. (CSE) Second Year, Erode Sengunthar Engineering College, Erode,Tamilnadu,India

††Faculty of Computer Science & Engg Dept, Erode Sengunthar Engineering College, Erode, Tamilnadu,India

Summary

In cloud environment, an enormous amount of data is produced everyday. Many organizations are now migrated to cloud and demand for resource is increasing. Hence the providers are now delivering multi-cloud environment to meet this demand. If multiple providers cooperatively work together the availability of resource can be increased. But still clients are worrying that their data is correctly stored and maintained by providers without intact. Though the providers are provide enough security there are still many security issues happening in cloud. Integrity verification of client data is made by using a technique called Provable Data Possession(PDP). This survey paper provides overview about various Provable Data Possession techniques in cloud..

Key words:

Integrity verification, Multi cloud, Provable Data Possession

1. Introduction

Cloud computing is a trend in the present day scenario with almost all the organizations are entering into it. Cloud computing is the collection of virtualized and scalable resources and provide service based on "pay only for use" strategy. It is a faster profit growth point by providing a comparably low-cost, scalable, position-independent platform for clients' data. It is constructed based on open architectures and has the capability to incorporate multiple internal and/or external cloud services together to provide high interoperability. Such a distributed cloud environment is called **Multi-Cloud.**

Cloud computing is different from information technology by,

1. Outsourced resources – This includes both hardware and software. On-site file server can provide a source for file handling, data storage, and information backup. In cloud computing also provide this service but the vendor foots the costs of these computer resources.

Pay-as-you-go – Cloud computing will require a basic start up fee followed by a monthly usage charge but it cost lower than installing On-site file management. User can pay charge based on cloud time consumption, operating space and additional software features.

On-demand – In cloud computing, user pay for what you use. It satisfy every known business computer need, not all features are used by every purchaser. Thus the purchaser is freed to pursue more profit-oriented activities.

CHARACTERISTICS:

As per NIST (National Institute of Standards and Technology)[1] cloud computing should possess this basic characteristics,

1. On-demand self-service - A consumer can unilaterally provision computing capabilities.

2. Broad network access - Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms.

3. Resource pooling - The provider's computing resources are pooled to serve multiple consumers, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

4. Rapid elasticity - Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in.

5. Measured service - Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service.

SERVICE MODELS:

Basic service models in cloud computing are,

Infrastructure as a Service (IaaS)

Iaas is a basic cloud service model where cloud providers offer computers as physical or as virtual machines. It also offers servers, storage, load balancers etc. It serves on a utility basis (i.e) cost reflect the amount of resources allocated and consumed.

Cloud Providers : IBM Blue Cloud, Joyent, GoGrid, SunGrid, Amazon EC2

2. Platform as a Service (PaaS)

In PaaS model, cloud providers deliver a computing platform like operating system, database and web server. The resources here scale automatically to match demand and user not have to allocate resources manually.

Platform as a Service: Google App Engine, Oracle Saas Platform, MS Azure.

Data: Amazon S3, Google Base, Amazon SimpleDB, Microsoft SSDS.

3. Software as a Service (SaaS)

In SaaS model, cloud providers install and operate application software in the cloud and cloud users access the software. The cloud users do not manage the cloud infrastructure and platform on which the application is running. It provide features like elasticity, virtual desktop, emulator etc.

SaaS: Salesforce, Google Docs, Facebook, LinkedIn, Doodle.

4. Cloud Clients

Cloud clients are users who access this service model based on their need. They can access resource through web browser, mobile apps, thin client etc.

CLOUD TYPES:

Public cloud

A large organization owns the cloud infrastructure and sells cloud services to industries or public. Public cloud applications, storage, and other resources are made available to the general public by a service provider. These services are free or offered on a pay-per-use model.

Community cloud

Several organizations that have similar polices, objectives, aims and concerns share the cloud infrastructure. The common concerns (security, compliance, jurisdiction, etc.), whether managed internally or by a third-party and hosted internally or externally.

Hybrid cloud

Hybrid cloud is a composition of two or more clouds (private, community or public) that remain unique entities but are bound together, offering the benefits of multiple deployment models. It enables data and application probability

Private cloud

The cloud infrastructure is owned or leased by a single organization and is operated solely for that organization Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third-party and hosted internally or externally.

CLOUD ARCHITECTURE:

It is a systems architecture involved in the delivery of cloud computing, involves multiple cloud components communicating with each other over a loose coupling mechanism such as a messaging queue. Elastic provision implies intelligence in the use of tight or loose coupling as applied to mechanisms such as these and others. It have four basic steps,

1.Cloud infrastructure (eg. Billing VM)

2.Cloud service (eg. Queue)
3.Cloud platform (eg. Web Frontend)
4.Cloud storage (eg. Database)

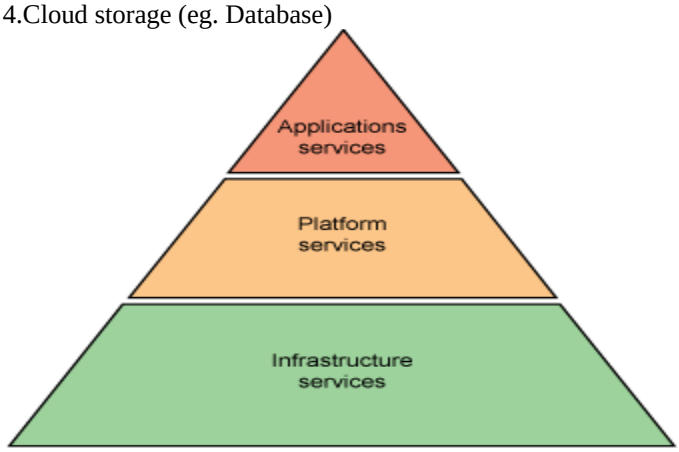

Fig 1. Cloud Architecture

DISTRIBUTED SERVERS:

The servers are not housed in the same location. Often, they are in geographically disparate location. But for cloud subscribers, it act as right next to each other. It gives service provider more flexibility in option and security. If any failure occurred at one site the service be still accessed from another site. If cloud need more hardware they need not throw more server in the safe room they simply add them at another site as a part of cloud.

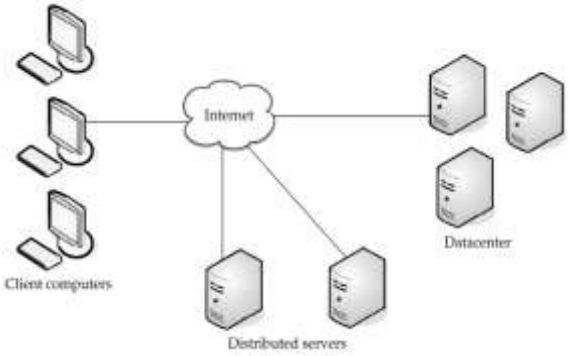

Fig 2. Distributed Servers

CLOUD ENTITIES:

1. End User - End Users need to access certain resources in the cloud and should be aware of access

agreements such as acceptable use or conflict of interest and concerns transmission integrity.

2. System Architect - System architects are employed with writing the policies that pertain to the installation and configuration of hardware components such as firewalls, servers, routers, and software.

3. Developers - Developers build an application in the cloud need to access the infrastructure where the development environment is located and improve development through elasticity of resources.

4. Third Party Auditors (TPA) - Third party auditors are used by clients and providers alike to determine the security of the cloud implementation. Depending on the level of commitment to security and usefulness a cloud vendor may choose to submit itself to regular security assessments in an attempt to obtain accreditation. The accreditation process needs to be undertaken every three years.

CLOUD COMPUTING SECURITY:

There are many security issues in cloud and they are classify into main issues,

1. Security issues faced by cloud providers
2 .Security issues faced by their customers

Basic Principles in Information Security:

1. Confidentiality – It is used to prevent disclosure of information to unauthorized person. It is necessary for maintaining the privacy of personal information.

2. Integrity – The data should not get modified without knowingly. The data should remain intact unless it is modified by authorized person.

3. Availability – The information must be available whenever it is needed. Ensure availability should always prevent DoS attacks.

CLOUD COMPUTING CONTROLS:

1. Detterent control – It prevent any purposeful attack in cloud. It's like warning sign but it does not reduce actual vulnerability of system.

2. Preventive control – It strengthen system by managing the vulnerabilities. It prevent attack by covering attacks and reduce damage to system.

3. Corrective control – It reduce the effect of attack. It take some corrective action when attack is happening.

4. Detective control – It detect that any attack have occurred in system. In case of any attack it signals either to corrective or preventive control.

DIMENSIONS OF CLOUD SECURITY:

- **Security and Privacy :**

1. Identity management - Cloud providers either integrate the customer's identity into their own infrastructure or provide an identity management solution of their own.

2. Physical and personnel security - Providers ensure that physical machines are secure and that access to these machines and relevant customer data is documented.

3. Availability - Cloud providers assure customers will have regular and predictable access to their data

4. Application security - Cloud providers ensure that applications available as a service in cloud are secure by implementing testing and acceptance procedures for outsourced code.

5. Privacy - Providers ensure that all critical data (eg. credit card numbers) are masked and that only authorized users have access to data in its entirety.

- **COMPLAIANCE:**

1. Business continuity and data recovery - Cloud providers have business continuity and data recovery plans to ensure that service can be maintained in case of a disaster or an emergency and that any data loss will be recovered.

2. Logs and audit trails - Cloud providers work with their customers to ensure that these logs and audit trails are properly secured, maintained for as long as the customer requires

3. Unique compliance requirements - The data centers maintained by cloud providers subject to compliance requirements and customer data may not remain on the same system, or in the same data center or even within the same provider's cloud

- **LEGAL ISSUES:**

Public records - Legal issues include records-keeping requirements in the public sector and are required by law to retain and make available electronic records .

The rest of this paper is organized as follows. In Section 2, the various related work that have been undergone is discussed. undergone in . In Section 3 the overview of various integrity verification and availability techniques of various paper is described. In Section 4 the comparative study of all techniques along with their advantage and disadvantage is tabulated. In section 5 the conclusion of this paper is given.

2. Related works

The availability of the outsourced data is increased by using technique called Scalia[2] which adopts the placement of data among multiple storage providers and optimize the cost. This approach have multiple datacenters based on access dependencies of user and can suitable for multi-cloud storage efficiently. High availability of data is provided by scheme called High-Availability and Integrity Layer (HAIL)[3] which is a distributed cryptographic system that permits a set of servers to prove to a client that a stored file is intact and retrievable. To check the integrity of outsourced data , researchers provides two basic approaches called Provable Data Possession (PDP) [4] and Proofs of Retrievability (POR) [5].

First approach proposed the PDP model[4] for ensuring possession of files on untrusted storages without downloading the actual data and provided an RSA-based scheme for a static case. It also includes a public verifiability, with this anyone, not just the owner can challenge the server for data possession. This extends the application areas of PDP protocol by separating the data owners and the users. But this is insecure against replay attacks in dynamic scenarios because of dependencies in index of data blocks. Moreover, they are not fit for multi-cloud storage because there is no homomorphism property in the verification process. To overcome static file storage limits in PDP and to provide dynamic data operations in PDP the Scalable PDP [6] model have been proposed. It is a lightweight PDP scheme based on cryptographic hash function and symmetric key encryption. But due to the lack of randomness in the challenges and by using previous metadata, the servers can deceive the owners.

The another drawback in this model is block size and its length are fixed and hence the modification of blocks cannot be done anywhere. Based on this work two Dynamic PDP[7] has been proposed. First is basic scheme, called DPDP-I and Second is the 'blockless' scheme, called DPDPII. But these schemes are also not effective for a multi-cloud environment .

Second approach is POR scheme[5], it describes the preprocessing steps that the clients should do before sending their file to a CSP. But this also not allow for updating the data efficiently. An improved version of this protocol called Compact POR[8] has been proposed. This technique which uses homomorphic property to aggregate a proof into authenticator value but their solution is also static and could not prevent the leakage of data blocks in the verification process. The dynamic scheme with cost by integrating the Compact POR scheme and Merkle Hash Tree (MHT) into the DPDP has been proposed. Several POR schemes and models have been recently proposed by using RAID techniques. It introduced a distributed cryptographic system that allows a set of servers to solve the PDP problem. It is based on an integrity protected error correcting code (IP-ECC), which improves the security and efficiency of existing tools. Here , a file must be transformed into distinct segments with the same length, which are distributed across servers. It is more suitable for RAID rather than a cloud storage.

3. Overview of Integrity Verification and Availability Techniques

Integrity verification should be made by clients to assure that their data has been properly stored and maintained in third party server. The availability and durability of data is provided by multi-cloud concept. The overview of various Provable Data Possession and Proof of Retrievability techniques are summarized as follows.

3.1 Scalia for efficient Multi-Cloud

Scalia[2], a cloud storage brokerage solution that continuously adapts the placement of data based on its access pattern and subject to optimization objectives, such as storage costs. It efficiently considers repositioning of only selected objects that may significantly lower the storage cost. The cost-effectiveness of Scalia against static placements and its proximity to the ideal data placement in various scenarios of data access patterns of available cloud storage solutions and of failures.

Scalia can run directly at the customer premises as an integrated hardware and software solution or can be deployed as a hosted service across several datacenters. It provides a scalable and highly available architecture with no single point of failure and able to guarantee higher availability than the storage providers.

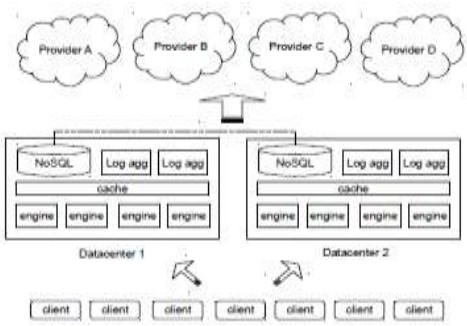

Fig 3. Multi – datacenter Architecture

3.2 Virtual Infrastructure Management in Private and Hybrid Clouds:

In Iaas, service provider's deploy data center as virtual machines[10]. With IaaS clouds growing popularity, tools and technologies are emerging that can transform an

organization's existing infrastructure into a private or hybrid cloud. To provide users with the same features found in commercial public clouds, private/hybrid cloud software must,

•Provide a uniform and homogeneous view of virtualized resources, regardless of virtualization platform

•Manage a VM's full life cycle, including setting up networks dynamically for groups of VMs and managing their storage requirements

• Support configurable resource allocation policies to meet the organization's specific goals

• Adapt to an organization's changing resource needs, even local resources are insufficient.

The challenge is in integrating multiple components to create complete IaaS cloud-building solutions. Private and hybrid clouds also face this challenge to efficiently manage the finite resources. However, existing VI managers rely on an immediate resource provisioning that implicitly assumes that capacity is practically infinite. Although this is a fair assumption for large cloud providers, but not applicable to smaller providers where the likelihood of being overloaded is greater.

3.3Collaborative Integrity Verification in Hybrid Clouds

In hybrid cloud an organization provides and manages some internal resources and the others provided externally. It may bring irretrievable losses to the clients due to lack of integrity verification[11] mechanism for distributed data outsourcing. Addressing the construction of a collaborative integrity verification mechanism in hybrid clouds, to support the scalable service and data migration, consider the existence of multiple cloud service providers to collaboratively store and maintain the clients' data. The verification is performed by a 5-move interactive proof protocol[12].

Different entities in hybrid cloud includes Cloud service providers (CSPs), who work together to provide data storage services and have enough storage spaces and computation resources; and Trusted third parties (TTPs), who are trusted to store the verification parameters[4] and offer the query services for these parameters.

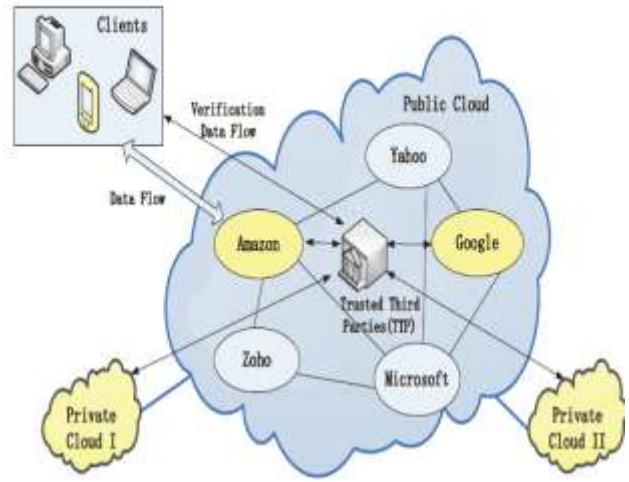

Fig 4.Integrity Verification in Hybrid Cloud

3.4 Dynamic Provable Data Possession

Proving the integrity of data stored at untrusted servers in resource-sharing networks is more important. In PDP[4], the client preprocesses the data and then sends it to an untrusted server for storage, while keeping a small amount of meta-data. The client later asks the server to prove that the stored data has not been tampered with or deleted ,without downloading the actual data. However, the original PDP scheme applies only to static files. The efficient constructions for dynamic provable data possession (DPDP I)[7], which extends the PDP model to support provable updates to stored data.

This use a new version of authenticated dictionaries based on rank information. It checks the integrity of file blocks. In (DPDP II) [7], Rank-based RSA trees are used which has a higher probability of detection, maintains logarithmic communication complexity but has increased update time.

3.5 Scalable and Efficient PDP

The efficient and secured outsourced data is addressed either by public key cryptography or requiring the client to outsource its data in encrypted form called E-PDP(Efficient-PDP)[13]. This technique is based entirely on symmetric key cryptography and not requiring any bulk encryption. It allows dynamic data, that efficiently supports operations, such as block modification, deletion and append. Two different approaches PDP[4] and POR[5] has been proposed. The POR is a public key-based technique allowing any verifier to query the server and obtain an interactive proof of data possession. This

property is called **public verifiability[9]**. The interaction can be repeated any number of times, each time resulting in a fresh proof.

The POR scheme uses special blocks (called sentinels) hidden among other blocks in the data. During the verification phase, the client asks for randomly picked sentinels and checks whether they are intact. If the server modifies or deletes parts of the data, then sentinels would also be affected with a certain probability.

3.6 Ensuring Data Storage Security in Cloud Computing

Ensuring cloud data storage security[14] for quality of service is an important aspect. To ensure the correctness, distributed scheme with the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving server(s) is introduced. The highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks. Erasure correcting code in the file distribution preparation provide redundancies and guarantee the data dependability.

By utilizing the homomorphic token with distributed verification of erasure-coded data, it achieves the storage correctness insurance as well as data error localization: whenever data corruption has been detected during the storage correctness verification, it can almost guarantee the simultaneous localization of data errors, i.e., the identification of the misbehaving server(s).

3.7 Secure Sensor Data Storage With Dynamic Integrity

Distributed data storage has gained increasing popularity for

its efficient and robust data management in wireless sensor networks (WSNs)[15]. But it makes challenge in building a highly secure and dependable yet lightweight data storage system. Sensor data are subject not only to Byzantine failures, but also to dynamic pollution attacks, as along as the adversary may modify/pollute the stored data by compromising individual sensor.

The resource-constrained nature of WSNs precludes the applicability of heavyweight security designs. Based on the principle of secret sharing and erasure coding, a hybrid share generation and distribution scheme to achieve reliable and fault-tolerant initial data storage by providing redundancy for original data components has been proposed. To dynamically ensure the integrity of the distributed data shares, an efficient data integrity verification scheme with algebraic signatures is proposed. It enables individual sensors to verify one protocol execution in all the pertaining data shares simultaneously in the absence of the original data. It have strong resistance against various attacks and are practical for WSNs.

The efficient and flexible dynamic data integrity checking scheme for verifying the consistency of data shares in a distributed manner. In this the data originating sensor partitions the original data into multiple shares based on the erasure coding and perfect secret sharing techniques. This reduces the communication and storage overhead as compared to the traditional replication-based techniques, and achieves reliable data storage by providing redundancy for original data components.

To ensure data integrity and availability, algebraic signatures with favorable algebraic properties, which allow the share holders to perform dynamic data integrity checks in a random way with minimum overhead. Since the data originating sensor appends a distinct parity block to each data share, all share holders can verify the distributed data shares independently in each check.

The false-negative probability can be reduced to almost zero, thus any unauthorized modifications can be detected in one verification operation and verify the integrity of aggregated data shares with great efficiency.

3.8 Space-Efficient Block Storage Integrity

It provide block-level integrity in encrypted storage systems[16], (i.e.),client detect the modification of data blocks in untrusted storage server. This neither change the block size nor the number of sectors accessed in modern storage systems. A trusted client component maintains state to authenticate blocks returned by the storage server for minimizing the size of state. Basic block integrity that exhibits a tradeoff between the level of security and the additional client's storage overhead, and evaluations requires an average of only 0.01 bytes per 1024-byte block. Due to the length-preserving requirements for cryptographic blocks, it is not possible to add information to each block (e.g., a MAC) to detect its modification.

In I/O applications, it is undesirable to put these MACs in separate blocks also stored at the service, which would require the retrieval of two blocks (one of data, one of MACs) on the critical path of client read operations. A trusted client component holds keys for encrypting blocks before their transmission to the storage service, and for decrypting blocks during their retrieval

along with integrity information.

3.9 Privacy-Preserving Audit

A number of online services, such as Google, Yahoo!, and Amazon, are starting to charge users for their storage. Customers often use these services to store valuable data such as email, family photos and videos, and disk backups. Today, a customer must entirely trust such external services to maintain the integrity of hosted data and return it intact. To make sure data integrity, protocols that allow a third party auditor to periodically verify the data stored by a service and assist in returning the data intact to the customer.

Most importantly, protocols are privacy preserving[17], and they never reveal the data contents to the auditor. To removes the burden of verification from the customer and to alleviates fear of both the customer's and storage service's about data leakage, it provides a method for independent arbitration of data retention contracts.

3.10 Compact Proof of Retrievability

Proof-of-Retrievability[5] system, is a data storage center which proves to a verifier it actually storing all of a client's data. But the challenge here is to build systems that are both e-client and provably secure that is, it should be possible to extract the client's data from any prover that passes a verification check. POR with full proofs of security against is a strongest model with two schemes.

1. Built from BLS signatures and secure in the random oracle model, has the shortest query and response of any proof-of-retrievability with public verifiability.

2. Builds elegantly on pseudorandom functions (PRFs) and is secure in the standard model, has the shortest response of any proof-of-retrievability scheme with private verifiability (but a longer query).

Both schemes rely on homomorphic properties to aggregate a proof into one small authenticator value. These two schemes provide a compact for retrievability (CPOR)[8].

3.11 Proofs of Retrievability.

The problem with remote storage is accountability. If remotely stored data is rarely accessed the users cannot be sure that their data is being stored honestly. If a remote storage provider experiences hardware failure and loses some data, it might reason that there is no need to notify its clients, since there is a good chance that the data will never be accessed and the client would never find out.

Alternatively, a malicious storage provider might even choose to delete rarely accessed files to save money. To assure such concerns, we would like a simple auditing procedure for clients to verify that their data is stored correctly. Such audits, called Proofs of Retrievability[5].

3.12 Proof of Retrievability via Hardness Amplification

Proofs of Retrievability (PoR)[5] allow the client to store a file F on an untrusted server, and later run an efficient audit protocol in which the server proves that it still possesses the client's data.

Constructions of PoR schemes attempt to minimize the client and server storage, the communication complexity of an audit, and even the number of file-blocks accessed by the server during the audit[18]. Different variants of the problem such as bounded-use vs. unbounded-use, knowledge-soundness vs. information-soundness provides optimal PoR schemes for each of these variants.

The literature survey of various security principles of availability, integrity and durability of data in cloud has been consolidated. The integrity verification principle has been recently verified by two popular approaches which is Provable Data Possession(PDP) and Proof of Retrievability(PoR). The various improvement in this technique is listed as literature survey.

4. Comparative Table

This comparative study provide consolidated techniques of various availability and integrity verification techniques along with their methodology used in each techniques. It also provide proof that each technique has been adopted to single or multi cloud.

The Comparative study of various techniques and the algorithm or implementation techniques that have adapted for each scheme is properly verified and tabulated in Table 1. The advantage and disadvantage of each techniques are also tabulated. As a result the study of this comparison table provide the importance of integrity verification that the client need to be done before storing their data to the third party server.

The availability of data is also greater importance and that can be done by integrating the several set of servers which is key concept in multi-cloud environment. The comparative study of various integrity and availability techniques that tabulated are as follows,

Table 1: Comparative survey of various availability and Integrity verification techniques

Scheme	Algorithm	Single/Multi Cloud	Description	Advantage	Disadvantage
Multi- datacenter	Scalia	Multi Cloud	Scalia method continuously adapts data in several storage provider and optimize the cost for clients.	1.High data availability 2.High data durability 3. minimize storage cost for clients	Latency overhead and scalability of prototype has not been described.
HAIL(High Availability and Integrity Layer)	RAID	Multi Cloud	Set of servers store clients data and give proof that stored file are intact and retrievable with high security.	1. High availability of data 2.Robust against active, mobile adversary.	RAID with POR protocol provide assurance only for static files and not for dynamic update of blocks.
PDP(Provable Data Possession)	HVT,E-PDP	Multi Cloud	It allows client to verify the server that possess their data without downloading the actual data by using Homomorphic Verifiable Tag.	1. It provides security to data based on RSA scheme. 2. It allows public verifiability in which access privilege can be set in cloud.	1. It is more efficient scheme but can applicable only for static files. 2. It is insecure against dynamic block of data.
SPDP (Scalable PDP)	PDP,MHT	Single Cloud	It provides secured data in encrypted form by using symmetric cryptographic key and also allows public verifiability.	1.It provides efficient PDP by encryption 2.It is light weight PDP scheme to support homomorphic hash function	1. It lacks in randomness hence by using the previous challenges, client can easily deceive the server.
DPDP-I (Dynamic PDP –I)	Authenticated Skip List	Single Cloud	It supports dynamic updates in each blocks which allows block modification .It use authenticated dictionaries based on rank list.	1.Block modification and updation of block is allowed. 2.Efficient integrity verification is made by querying and updating DPDP scenario.	It provides efficient verification but construction of rank based scheme is difficult
DPDP-II (Dynamic PDP – II)	RSA trees	Single Cloud	It also supports dynamic updation of data with blockless verification scheme in which entire data need not to be download.	1.Blockless verification where particular block can be queried for integrity verification. 2.RSA trees use homomorphic tag where tag are small and easy to use.	DPDP scheme with RSA tree construction is efficient with dynamic option but it cannot be adapt to the multi-cloud.
POR (Proof of retrievability)	MAC	Multi Cloud	It describes the preprocessing steps client should do before uploading the data to provider server by using Message Authentication Code.	1.Preprocessing steps can be made by client before storing their data. 2.It is the simple way to audit the server.	It is difficult to build the system for e-client probably with secured data during the audit
CPOR (Compact POR)	HVR	Multi Cloud	This technique which uses homomorphic property to aggregate a proof into authenticator. It gives dynamic cost by integrating the Compact POR scheme and Merkle Hash Tree (MHT) into the DPDP.	1.Homomorphic Verifiable Response from provider gives proof of stored data intact. 2.Dynamic cost of data provides more flexibility to user.	It provide authenticated proof value but their solution is also static and could not prevent the leakage of data blocks during the verification process.

4. Conclusion

Cloud is designed to provide a service to the external users. To compensate their needs the resources should be highly available. In this survey, it gives overview about cloud availability and various integrity verification techniques. In addition, comparative study of various availability and integrity verification schemes and its methodology are classified along with their adaptation to single/multi cloud environment.

References

[1] Peter Mell and Tim Grance(2009), "The NIST Definition of Cloud Computing" National Institute of Standards and Technology (NIST), Information Technology Laboratory (www.csrc.nist.gov)

[2] Thanasis G. Papaioannou, Nicolas Bonvin and Karl Aberer,(2012) "Scalia: An Adaptive Scheme for Efficient Multi-Cloud Storage" in International Conference for high performance computing and networking.

[3] Bowers K. D, Juels .A, and Oprea .A(2009), "Hail: a high-availability and integrity layer for cloud storage," in ACM Conference on Computer and Communications Security, pp. 187–198.

[4] Ateniese.G, Burns.R.C, Curtmola.R, Herring.J, Kissner.L, Peterson.Z.N.J, and Song. D.X (2007), "Provable data possession at untrusted stores," in ACM Conference on Computer and Communications Security, P. Ning, pp. 598–609.

[5] Juels.A and Kalisk.B (2007), "Pors: proofs of retrievability for large files," in ACM Conference on Computer and Communications Security, P. Ning, pp. 584–597.

[6] Ateniese.G, Pietro.R.D, Mancini.L.V, and Tsudik.G (2008), "Scalable and efficient provable data possession," in 4th international conference on Security and privacy in communication netowrks, pp. 1–10.

[7] Erway.C.C, Kupcu.A, Papamanthou.C, and Tamassia.R(2009), "Dynamic provable data possession," in ACM Conference on Computer and Communications Security, E. Al-Shaer, S. Jha, and A. D. Keromytis, Eds. ACM, pp. 213–222.

[8] Shacham.H and Waters.B (2008), "Compact proofs of retrievability," in ASIACRYPT, ser.Lecture Notes in Computer Science, Ed., vol. 5350. Springer,pp. 90– 107.

[9] Wang.Q, Wang.C, Li.J, Ren.K, and Lou.W (2009), "Enabling public verifiability and data dynamics for storage security in cloud computing," in ESORICS, ser. Lecture Notes in Computer Science, vol. 5789. Springer, pp. 355–370.

[10] Sotomayor.B. Montero.R.S. Llorente.I.M. and Foster.I.T(2009),"Virtual infrastructure management in private and hybrid clouds," IEEE Internet Computing, vol. 13, no. 5, pp. 14–22.

[11] Zhu.Y, Hu.H, Ahn.G.J, Han.Y, and Chen.S(2011), "Collaborative integrity verification in hybrid clouds," in IEEE Conference on the 7th International Conference on Collaborative Computing, October 15-18,pp. 197–206.

[12] Fortnow.L, Rompel.J, and Sipser.M (1988), "On the power of multi-prover interactive protocols," in Theoretical Computer Science, pp. 156–161.

[13] Zhu.Y, Wang.H, Hu.Z, Ahn.G.J, Hu.H, and Yau.S.S(2011), "Dynamic audit services for integrity verification of outsourced storages in clouds," in *SAC*, W. C. Chu, W. E. Wong, M. J. Palakal, and C.-C. Hung, Eds. ACM, pp. 1550–1557.

[14] Cong Wang, Qian Wang, and Kui Ren (2009), "Ensuring Data Storage Security in Cloud Computing", INTERNATIONAL workshop on QoS pp.1-9.

[15] Wang.Q, Ren.K, Lou.W, and Zhang.Y (2009), "Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance," Proc. of IEEE INFOCOM.

[16] Alina Oprea Michael Reitery Ke Yangz.*K* (2007)*,"*Space-Efficient Block Storage Integrity".

[17] Mehul A. Shah Ram Swaminathan Mary Baker (2008),"Privacy-Preserving Audit and Extraction of Digital Contents", HP Labs Technical Report No. HPL-32.

[18] Dodis.Y, Vadhan.S.P, and Wichs.D (2009), "Proofs of retrievability via hardness amplification," in *TCC*, ser. Lecture Notes in Computer Science, O. Reingold, Ed., vol. 5444. Springer, pp. 109–127.