International Journal of Recent Advances in Multidisciplinary Topics
Volume 2, Issue 4, April 2021
https://www.ijramt.com | ISSN (Online): 2582-7839

Maintaining Security of the Data Over Cloud: A Review

S. R. Dahake[1*], E. M. Chirchi[2]
[1,2]*Department of Computer Science Engineering, Shreeyash College of Engineering and Technology, Aurangabad, India*

***Abstract*: Recently, cloud computing is the fastest growing technology in the IT field. Cloud computing provides us a means by which we can access the applications as utilities, over the Internet. It allows us to create, configure, and customize applications online. The term cloud refers to a network or internet. In other words, we can say that cloud is something, which is present at remote location. Cloud can provide services over the network, i.e., on public networks or on private networks, e.g. WAN, LAN or VPN. Applications such as e-mail, web conferencing, Customer Relationship Management (CRM), and all run in cloud. Cloud computing refers to manipulating, configuring, and accessing the applications online. It offers online data storages, infrastructures and applications. To maintain the security of the data over cloud is studied in the present work.**

***Keywords*: Third Party Auditor (TPA), LAN, Customer Relationship Management (CRM).**

1. Introduction

There are certain services and models working behind the scene making the cloud computing feasible and accessible to end users. The working models for cloud computing as follows:

i. Deployment models
ii. Service models

1) Deployment models

Deployment models define the type of access to the cloud, i.e., how the cloud is located? Cloud can have any of the four types of access as follows:

a) Public cloud
b) Private cloud
c) Hybrid cloud and
d) Community cloud

2) Service models

Service models are the reference models on which the cloud computing is based. These can be categorized into three basic service models as listed below:

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

There are many other service models all of which can take the form like XaaS, i.e., anything as a Service. This can be Network as a Service, Business as a Service, Identity as a Service, Database as a Service or Strategy as a Service. The Infrastructure as a Service (IaaS) is the most basic level of service.

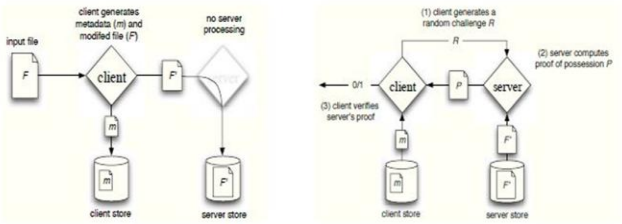
2. Literature Review

A teniese et. al. [1] investigated the provable data possession at untrusted stores. They suggested the random sampling of outsourced data and used RSA scheme with homomorphic linear authenticators for auditing the outsourced data. As it was the first scheme, privacy protection of data was not considered by them; hence the data was accessible by external auditor for auditing purpose. They introduced a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely-distributed storage systems. They presented two provably-secure PDP schemes that are more efficient than previous solutions, even when compared with schemes that achieve weaker guarantees. In particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data. Experiments using our implementation verify the practicality of PDP and reveal that the performance of PDP is bounded by disk I/O and not by cryptographic computation. They formally defined protocols for provable data possession (PDP) that provide probabilistic proof that a third party stores a file. They introduced the first provably-secure and practical PDP schemes that guarantee data possession. Their PDP schemes provide data format independence, which is a relevant feature in practical deployments and put no restriction on the number of times the client can challenge the server to prove data possession.

They focused on the problem of verifying if an untrusted server stores a client's data and introduced a model for provable data possession, in which it is desirable to minimize the file block accesses, the computation on the server, and the client-server communication. PDP incur a low (or even constant) overhead at the server and require a small, constant amount of

*Corresponding author: pnbsaurabh123@gmail.com

communication per challenge. Key components of the schemes were the homo-morphic verifiable tags. They allow verifying data possession without having access to the actual data file.



(a) Pre-process and store (b) Verify server possession

Fig. 1. Provable data possession

According to Wang et. al. [2], using cloud storage, users can remotely store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in cloud computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public audit ability for cloud storage is of critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. They proposed a secure cloud storage system supporting privacy-preserving public auditing. They enabled the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient. Hence, they proposed a system that will be using public key based homomorphic authenticator along with random masking technique to facilitate privacy of user data and enable public auditing of the user's data stored in cloud. They proposed to use the public key based homomorphic authenticator through which the TPA can perform the auditing task without asking for the copy of user's data and then integrating it with random masking technique. The protocol promises that TPA will not have necessary information to rebuild the correct group of linear equations and therefore could not have any knowledge about user's data that is stored in the cloud server during the auditing process. They also proposed that the file should be encrypted at user's side and the key is stored with the user before uploading the file to cloud. They assessed the performance of the proposed privacy-preserving public auditing schemes to show that they are indeed lightweight. They focused on the cost of the efficiency of the privacy-preserving protocol and the batch auditing technique.

Based upon above research, they proposed a privacy-preserving public auditing system for data storage security in Cloud Computing. They utilized the homomorphic linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy-preserving public auditing protocol into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency.

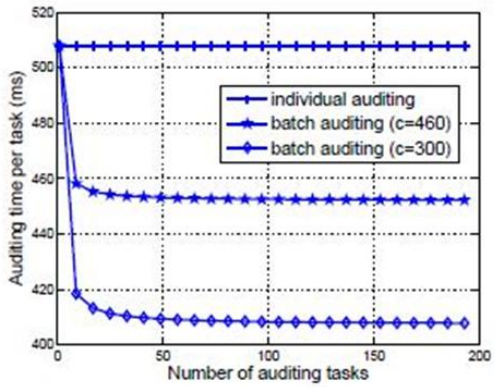


Fig. 2. Comparison on auditing time between batch and individual auditing. Per task auditing time denotes the total auditing time divided by the number of tasks

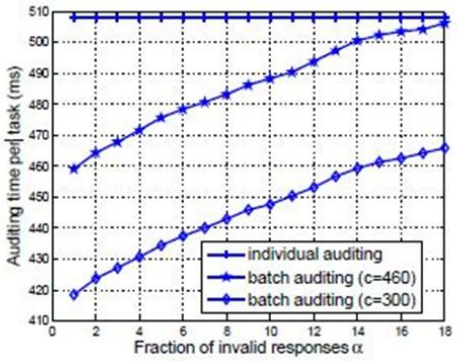


Fig. 3. Comparison on auditing time between batch and individual auditing, when-fraction of 256 responses are invalid

Shacham and Waters [3] given a proof-of-retrievability system, a data storage center must prove to a verifier that he is actually storing all of a client's data. The central challenge is to build systems that are both efficient and provably secure that is, it should be possible to extract the client's data from any prover that passes a verification check. In this paper, they gave the first proof-of-retrievability schemes with full proofs of security against arbitrary adversaries in the strongest model. Their first scheme built from BLS signatures and secures in the random oracle model feature sa proof-of-retrievability protocol in which the client's query and server's response are both extremely short. This scheme allows public verifiability: anyone can act as a verifier, not just the file owner. Our second scheme, which builds on pseudorandom functions (PRFs) and is secure in the standard model, allows only private verification. It features a proof-of-retrievability protocol with an even

shorter server's response than our first scheme, but the client's query is long. Both schemes rely on homomorphic properties to aggregate a proof into one small authenticator value.

They provided two contributions:

1. They described two new short, efficient homomorphic authenticators. The first, based on PRFs, gives a proof-of-retrievability scheme secure in the standard model. The second, based on BLS signatures, gives a proof-of-retrievability scheme with public verifiability secure in the random oracle model.
2. Their schemes are the first with a security proof against arbitrary adversaries in this model. The scheme with public retrievability features a proof-of-retrievability protocol in which the client's query and server's response are both extremely short: 20 bytes and 40 bytes, respectively, at the 80- bit security level. The scheme with private retrievability features a proof-of-retrievability protocol with an even shorter server's response than our first scheme: 20 bytes at the 80-bit security level, albeit at the cost of a longer query.

With cloud storage services, it is commonplace for data to be not only stored in the cloud, but also shared across multiple users. However, public auditing for such shared data — while preserving identity privacy — remains to be an open challenge presented by Wang et. al. [4]. In their investigations, they proposed the first privacy-preserving mechanism that allows public auditing on shared data stored in the cloud. In particular, they exploited ring signatures to compute the verification information needed to audit the integrity of shared data. With our mechanism, the identity of the signer on each block in shared data is kept private from a third party auditor (TPA), who is still able to publicly verify the integrity of shared data without retrieving the entire file. Our experimental results demonstrate the effectiveness and efficiency of our proposed mechanism when auditing shared data. They concluded as, the first privacy preserving public auditing mechanism for shared data in the cloud. They utilized ring signatures to construct homomorphic authenticators, so the TPA is able to audit the integrity of shared data, yet cannot distinguish who is the signer on each block, which can achieve identity privacy. To improve the efficiency of verification for multiple auditing tasks, we further extend our mechanism to support batch auditing. An interesting problem in our future work is how to efficiently audit the integrity of shared data with dynamic groups while still preserving the identity of the signer on each block from the third party auditor.

Wang and Ren [5] came up with an idea towards publicly auditable secure cloud data storage services. Cloud computing is the long dreamed vision of computing as a utility, where data owners can remotely store their data in the cloud to enjoy on-demand high-quality applications and services from a shared pool of configurable computing resources. While data outsourcing relieves the owners of the burden of local data storage and maintenance, it also eliminates their physical control of storage dependability and security, which traditionally has been expected by both enterprises and individuals with high service-level requirements. In order to facilitate rapid deployment of cloud data storage service and regain security assurances with outsourced data dependability, efficient methods that enable on-demand data correctness verification on behalf of cloud data owners have to be designed. In this paper they proposed that publicly auditable cloud data storage is able to help this nascent cloud economy become fully established. With public auditability, a trusted entity with expertise and capabilities data owners do not possess can be delegated as an external audit party to assess the risk of outsourced data when needed. Such an auditing service not only helps save data owners' computation resources but also provides a transparent yet cost-effective method for data owners to gain trust in the cloud. We describe approaches and system requirements that should be brought into consideration, and outline challenges that need to be resolved for such a publicly auditable secure cloud storage service to become a reality.

They also stated that cloud computing has been envisioned as the next-generation architecture of enterprise IT. In contrast to traditional enterprise IT solutions, where the IT services are under proper physical, logical, and personnel controls, cloud computing moves the application software and databases to servers in large data centers on the Internet, where the management of the data and services are not fully trustworthy. This unique attribute raises many new security challenges in areas such as software and data security, recovery, and privacy, as well as legal issues in areas such as regulatory compliance and auditing, all of which have not been well understood. In this article we focus on cloud data storage security. They first presented network architecture for effectively describing, developing, and evaluating secure data storage problems. We then suggested a set of systematically and cryptographically desirable properties for public auditing services of dependable cloud data storage security to become a reality. Through in-depth analysis, some existing data storage security building blocks are examined. The pros and cons of their practical implications in the context of cloud computing are summarized. Further challenging issues for public auditing services that need to be focused on are discussed too. They believed security in cloud computing, an area full of challenges and of paramount importance, is still in its infancy now but will attract enormous amounts of research effort for many years to come.

Prasanthi et. al. [6] provided an efficient auditing protocol for secure data storage in cloud computing. Computing is a type of distributed computing whereby resources and applications are shared over the internet. These applications are stored in one location and can be accessed in different location by any authorized users where the user does not need any infrastructure. In cloud storage, while outsourcing trust worthiness of the data is a scary task in cloud. To ensure the integrity of dynamic data stored in the cloud, external Third Party Auditor (TPA) is acquainted in a cloud infrastructure. For enabling public auditing in cloud data storage security, users can resort to an external auditor to check integrity of an outsourced data. The third party auditor (TPA) should meet the following fundamental requirements:

1) TPA should be able to efficiently audit the cloud data without revealing the original data, and it should not add burden to the cloud user;

2) Auditing process should not bring new vulnerabilities towards the user data.
3) Integrity of the data is protected against TPA by invoking some cryptographic techniques to ensure the storage correctness in cloud.

In particular, this scheme achieves batch auditing where multiple delegated auditing tasks from different users, can be performed by the TPA and further enables TPA to perform data dynamics operations. Thus, the performance analysis depicts that the proposed schemes are more sheltered and highly competent. They investigated the problem of data integrity in cloud data storage, which is essentially a distributed storage system. It involves the hashing technique to achieve the correctness of data over cloud server. Then propose an effective and flexible distributed scheme with explicit dynamic data support, including block update, delete, and append. To support efficient handling of multiple auditing tasks, to further explore the technique of bilinear aggregate signature to extend the main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously. Extensive security and performance analysis show that the proposed scheme is highly efficient and provably secure.

Shah and Baker [7] stated that a growing 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. Unfortunately, no service is infallible. To make storage services accountable for data loss, they presented 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, our protocols are privacy-preserving, in that they never reveal the data contents to the auditor. Our solution removes the burden of verification from the customer, alleviates both the customer's and storage service's fear of data leakage, and provides a method for independent arbitration of data retention contracts.

Their protocols have three important operations, initialization, audit, and extraction, and they primarily focused on the latter two. For audits, the auditor interacts with the service to check that the stored data is intact. For extraction, the auditor interacts with the service and customer to check that the data is intact and return it to the customer. These protocols have the following salient features:

1) Audit: With minimal long-term state, an auditor can efficiently and repeatedly check stored contents on behalf of the customer. In these audits, the service must prove that contents are completely unchanged.
2) Extraction: Upon retrieval, if the customer doubts the integrity of the data, the customer can use the extraction protocol which routes the data through the auditor to the customer.

During extraction, the auditor can determine which party is at fault: whether the service lost data or which party is cheating by not obeying the protocol. Thus, the auditor can arbitrate data retention contract.

1) All our protocols do not reveal the data contents to the auditor. Our auditing protocols are zero-knowledge, providing no added information to the auditor. Their extraction protocols prevent an adversarial auditor from recovering the data contents. Yet, they still allow the auditor to check the integrity of retrieved data and forward it so that a customer can efficiently recover the contents.
2) A customer does not need to maintain any long-term state. For example, he does not need to keep "fingerprints" or hashes to audit the stored data, or keep secret keys to decrypt the stored data upon retrieval.

A straight forward solution for maintaining privacy during audits is for the customer to encrypt his contents using symmetric-key encryption and keep those keys intact and secret from uninvited parties. Then, the auditor can use existing provably secure, challenge-response schemes on the encrypted contents. This solution is unsatisfactory because an unsophisticated customer is increasingly likely over time either to lose the keys and be unable to recover the contents, or to leak the keys.

In their protocols we shifted the burden of keeping these secret keys to a storage service. Since services are already in the business of maintaining customers' data and privacy, the keys are safer with them. Keeping the data content private from the service is optional. A customer can keep the keys and encrypted data with the same service, thereby revealing the contents to that service and allowing it to provide value-added features beyond storage like search. Otherwise, the customer can separate the keys and encrypted data onto non-colluding services to maintain complete privacy. The auditor is responsible for auditing and extracting both the encrypted data and the secret keys. Their protocols, however, never reveal the secret key to the auditor. Although they presented the protocols for handling the encrypted data for completeness, they are straight forward extensions of existing techniques. Their main contributions are as follows:

1) In motivating and specifying the problem of privacy-preserving audit and extraction of digital data and
2) Privacy-preserving protocols for auditing and extracting the encryption key.

In this article they presented these protocols and show how they provably ensure data integrity and data privacy from the auditor to support their protocols, storage services must export hooks for challenge-response queries and compute expensive functions for responses. To avoid these overheads, we can batch many files together into a single file and check that single file all at once. Since their protocols mostly send small hashes and auditing keys requires at most two exponentiations, the main overhead comes from computing HMACs over the entire contents. If we limit the number of checks, however, this overhead can be tolerable for a service. Also, they measured the performance of a SHA-1 HMAC over files stored on five 500GB SATA disks with a 2-core 2GHz Intel Xeon 5130 at 362 MB/s. At this peak rate, 50 CPUs in parallel can check 1PB in 16 hours. Spread over 30 days, this work imposes less than 2% overhead. Since many large-scale storage services handle an archival workload in which most data is rarely touched,

checking monthly is reasonable.

For future work, we should consider modifying previous schemes to allow privacy preserving audit and extraction. Certainly a hybrid as mentioned above is simple, but it still imposes the encryption and decryption overheads that the storage service and customer experience with their protocols. It may be possible to eliminate the encryption key altogether. To do so, we must first extend the formal definitions of proofs of possession or proofs of retrievability to include a notion of privacy from the auditor.

Their current schemes require the auditor to be trusted and not collude with either party. In real world auditing situations in other contexts (e.g. financial auditing), such collusions can occur and have occurred in the past. In this case, a scheme that allows auditing by multiple auditors without the traditional overheads of Byzantine fault-tolerance techniques would be useful. They described methods for privacy preserving auditing and extraction of digital contents. Their two main contributions are

1) The motivation, assumptions, and setting for third-party privacy preserving auditing and extraction, and
2) Various privacy-preserving protocols for auditing and extraction of data stored with a service provider. Auditing involves a third-party auditor that remotely verifies that the stored data are intact. For extraction, the auditor verifies the data is intact and returns it to the customer, ensuring that he received the original data. Thus, with these schemes, an auditor can arbitrate data retention contracts between storage provider and customer.

These schemes divide the data into two pieces, an encryption key and the encrypted data. These protocols allow an auditor, with minimal long-term state, to audit both those pieces and extract those pieces without revealing the underlying contents of either. Using our protocols, all these properties can be achieved without requiring the customer to maintain any long-term state (secret keys or hashes). The protocols for the encrypted data rely on cryptographic hashes and symmetric key encryption. We present protocols for the encryption key that have varying assumptions, but all assume that the computing the discrete log is difficult. We believe protocols like ours will be necessary if outsourcing is truly to catch-on for preserving digital contents.

Srijanya and Kasiviswanath [8] came up with this. Cloud computing gets its name as a metaphor for the internet. It is the next generation platform to provide resources as the services to the end users. In cloud storage system, the clients store their data in the server without keeping a local copy. Clients store their data in the private clouds but when storage expansion is needed they move to public clouds. Security is the major concern in the public clouds. The security mechanisms for private and public clouds are different. It may be possible that an unauthorized user can access the data from the public clouds. Hence, it is of critical importance that the client should be able to verify the integrity of the data stored in the remote un-trusted server. There may be security services offered by public clouds but they are not sufficient. In order to address the security issues, Trusted Third Party Auditing (TPA) is used as a service for private and public clouds, which offers various services to check for the integrity of the data. TPA mechanisms offer various auditing mechanisms such as read, write, update to verify the integrity of the data stored in the public clouds. They presented such an auditing model based on Merkle Hash Tree. In this work they conducted a study on possible auditing mechanisms which can be offers as a service over hybrid/public clouds. Such services can be subscribed by the users to verify the integrity of the data stored in the public clouds.

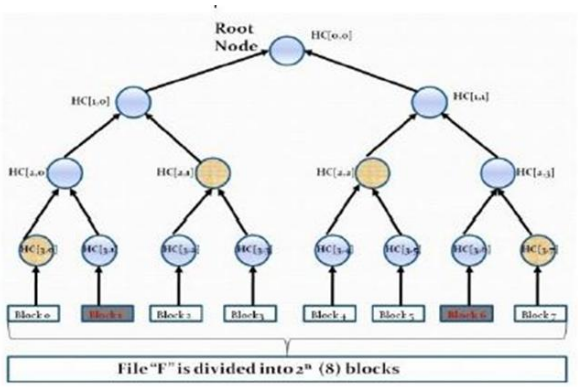


Fig. 4. Merkle hash tree

Merkle's construction built on the technique of Lamport's one-time signatures. Such a signature requires a setup stage in which many secret values are selected and the results of applying a hash function to each of them are published. To be concrete, we think of the hash function as SHA1, which outputs a 160-bit value. In its simplest form, the setup procedure for a one-time signature generates 160 secret values xi and 160 secret values (they used 160 here since SHA1 outputs a 160-bit value), and the sequence of pairs HASH(xi) and HASH is made public. To sign a message m, the signer first computes the 160-bit hash or message digest of m, md = HASH (m). For each of the 160 bits of md, the signer then reveals xi if the i'th bit of md is 0 and otherwise. An adversary can't forge such a signature, unless he can invert the hash function HASH. However, each sequence of 320 published HASH values can only be used once. Thus, the storage associated with the original one-time signature scheme grows too large to be practical for general use Merkle proposed a method to sign multiple messages without the enormous cost of storing two secret values per bit to be signed. His construction makes use of a binary tree, where each node is associated with a bit-string. The bit strings associated to each leaf are the hash of a secret value associated to that leaf, and each internal node of the tree is assigned HASH(L‖R), where L‖R represents the concatenation of the values assigned to the left and right child nodes. Rather than randomly generating and storing the secret values for each leaf, the i'th secret value can be determined from a pseudo-random generator as PRNG(secret,i). The root of the tree is made public. This key generation process is time consuming, but is a one-time cost.

A putative secret leaf value can be "authenticated" with respect to the root. Although each node value is considered to be public, the Verifier only needs to know the root value; the Prover supplies the additional public information to the

Verifier: namely the values of all of the siblings of the nodes on the path to the root. The Verifier can compute the hash of the secret leaf value, then hash the result together with its sibling's value, etc., all the way up to the root. The secret value is accepted as genuine if the final value matches the published root value. The Merkle-Tree traversal problem can be thought of as the problem of easing the Prover's burden of "reminding" the verifier of the node values. Realizing that neither storing all node values (exponential space cost in the height of the tree) nor recalculating the node values on the fly (up to exponential time cost) would be an efficient use of resources, Merkle found a way to amortize the cost of recalculating the required nodes over multiple "rounds" (1 round = output required for 1 leaf verification). For a tree of height H, Merkle's scheduling algorithm required only O(H) HASH evaluations per round, and space to store O(H^2) intermediate hash values. For medium-size trees this original algorithm already embodied a reasonable degree of storage and computation efficiency.

They presented a model for hybrid cloud in Remote sensing data using Amazon S3, Open Stack Swift and designing TPA mechanism which can be used as service and data is stored safely. Currently, the TPA and private cloud existing as service, same thing can be extended to public clouds the feasibility can be studied on the SLA's.

Srinivas [9] presented privacy-preserving public auditing in cloud storage security. The Cloud Computing is the new vision of computing utility, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or cloud provider interaction. Cloud computing technologies can be implemented in a wide variety of architectures, under other technologies and software design approaches. The security challenges cloud computing presents the burden of local data storage and maintenance. Public auditability for cloud data storage security is of critical importance so that users can resort to an external audit party to check the integrity of outsourced data when needed. To securely introduce an effective third party auditor to check the integrity of outsourced data. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. In this investigation, he proposed a secure cloud storage system supporting privacy-preserving public auditing. We utilize the homomorphic non-linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also all evirates the users' fear of their outsourced data leakage.

They proposed a privacy-preserving public auditing system for data storage security in cloud computing. We utilize the homomorphic non-linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data security. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy-preserving public auditing protocol into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that these schemes are probably secure and highly efficient.

Krishna and Rama [10] came up with this idea. Cloud computing is the arising technology to minimize the utilizer burden in the updating of data in business utilizing internet. Instead of local data storage and maintenance, the utilizer is availed with the cloud storage so that the utilizer can remotely store their data and relish the on-demand high quality application from a shared pool of resources. The data stored must be forfended in the cloud storage. The security challenges cloud computing presents the encumbrance of local data storage and maintenance. Public auditability for cloud data storage security is of critical paramountcy so that users can resort to an external audit party to check the integrity of outsourced data when needed. To securely introduce an efficacious third party auditor to check the integrity of outsourced data. To securely introduce an efficacious TPA, the auditing process should bring in no incipient susceptibilities towards utilizer data privacy, and introduce no supplemental online burden to utilizer. They proposed a secure cloud storage system fortifying privacy-preserving public auditing. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient.

Also, they proposed that to fully ensure the data integrity and save the cloud users' computation resources as well as online burden, it is of major importance to approve public auditing service for cloud computing information storage, so that users may resort to a unique Third Party Auditor (TPA) to audit the outsourced data whenever we needed. The third party auditing, who has to know more and capabilities that users do not, can sequentially check the integrity of all the information stored in the cloud on behalf of the users, which provides an easier way for the users to ensure their storage correctness in the cloud computing. Moreover, in addition to help users to evaluate the risk of their subscribed cloud data services, the audit result from TPA would also be beneficial for the cloud service providers to improve their cloud based service platform, and even serve for independent arbitration purposes. In a word, enabling public auditing services will play an important role for this nascent cloud economy to become fully established; where users will need ways to assess risk and gain trust in the cloud.

1) They motivated the public auditing system of data storage security in cloud computing and provide a privacy preserving auditing protocol. Our scheme enables an external auditor to audit user's cloud data without learning the data content.
2) Their scheme was the first to support scalable and efficient privacy preserving public storage auditing in Cloud. Specifically, our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA in a privacy preserving manner.

3) They proved the security and justify the performance of our proposed schemes through concrete experiments and comparisons with the state-of-the-art.

Concluding remarks of their research are, they proposed a protecting privacy from public auditing through security storage in cloud computing. To fully ensure the data integrity and save the cloud users' computation resources as well as online burden, they also explored a cloud computing new entity privacy preserving public auditing system for the purpose of data storage security, where TPA works on auditing details without need of data which was stored locally. Here we use the authenticator with feature of homomorphism and also using technique random mask to create trust on cloud that used TPA will not get or bother about the information which was stored by the user while auditing process, it also reduces the workflow to cloud user from the annoying and cost efficient auditing task, but also take the edge off the users to decrease the fear of their uploaded data privacy.

Under taking TPA may concurrently handle different audit levels from various users for their updated data files, in addition we extend our privacy-preserving public auditing protocol from single user to multi-user, here TPA workouts on various number of auditing tasks parallel. Efficient security and performance analysis gives reports that the proposed techniques are secure and highly efficient. The mighty features of the proposed schemes reduce the burden of economies in future for Cloud Computing.

Goyal [11] given the concept of cloud computing is one of the major theories in the world of IT. Its services are now being applied to several IT scenarios. Cloud Computing is the internet based computing which provides users with a number of services. Users store their data in the cloud without the burden of local data storage. As the user no longer have physical possession of data so the integrity and security of data become the major concern in the cloud computing. Data stored on the cloud server may be get corrupted and sometimes even the cloud service provider for his own benefit like for more space on data center can discard the user data which is not used for a longer time. In order to maintain the integrity of data, the user takes the assistance of a Third Party Auditor (TPA) checks the integrity of data on user demand and the released audit reports help the user to evaluate the risk of their services. TPA have an experience that user does not have and have capability to check integrity of data which is not easy for the user to check. This paper highlighted the basics of cloud computing, general model and different approaches used for TPA.

Cloud computing provides many benefits to their user but security is major issues in cloud computing. As user store their data to cloud data centers but as user does not know the exact location of their data so integrity of data is very important. To check the integrity of data there are many solutions available. One of solution is to take the assistance of a third party auditor. Different authors provide different solutions for implementing third party auditor. Each scheme has its own merits and demerits.

Paigude and Chavan [12] presented the cloud computing is a latest technology which provides various services through internet. The Cloud server allows user to store their data on a cloud without worrying about correctness and integrity of data. Cloud data storage has many advantages over local data storage. User can upload their data on cloud and can access those data anytime anywhere without any additional burden. The User doesn't have to worry about storage and maintenance of cloud data. But as data is stored at the remote place how users will get the confirmation about stored data. Hence Cloud data storage should have some mechanism which will specify storage correctness and integrity of data stored on a cloud. The major problem of cloud data storage is security. Many researchers have proposed their work or new algorithms to achieve security or to resolve this security problem.

In their study, they suggested a new innovative idea for Privacy Preserving Public Auditing with watermarking for data Storage security in cloud computing. It supports data dynamics where the user can perform various operations on data like insert, update and delete as well as batch auditing where multiple user requests for storage correctness will be handled simultaneously which reduce communication and computing cost. Wang has given a public auditing allows TPA along with user to check the integrity of the outsourced data stored on a cloud and Privacy Preserving allows TPA to do auditing without requesting for local copy of the data. Through this scheme, TPA can audit the data and cloud data privacy is maintained. It contains 4 algorithms as follows:

1) *Keygen*: It is a key generation algorithm used by the user to setup the scheme.
2) *Singen*: It is used by the user to generate verification metadata which may include digital signature.
3) *GenProof*: It is used by CS to generate a proof of data storage correctness.
4) *Verifyproof*: Used by TPA to audit the proofs.

It is divided mainly into two parts as setup phase and audit phase.

1) Setup Phase: Public and secret parameters are initialized by using Keygen and data files f are preprocesses by using Singen to generate verification metadata at CS and delete its local copy. In preprocessing user can alter data files F.
2) Audit Phase: TPA issues an audit message to CS. The CS will derive a response message by executing Gen proof. TPA verifies the response using F and its verification metadata.

TPA is stateless i.e. no need to maintain or update the state information of audit phase. Public key based homomorphic linear authentication with random masking technique is used to achieve privacy preserving public auditing. TPA checks the integrity of the outsourced data stored on a cloud without accessing actual contents. Existing research work of proof of retrievability (PoR) or Proofs of Data Possession (PDP) technique doesn't consider data privacy problem. PDP scheme first proposed by A teniese et. al. used to detect large amount corruption in outsourced data. It uses RSA based Homomorphic authentication for auditing the cloud data and randomly sampling a few blocks of files. A Second technique proposed by Juels as Proofs of Retrievability (PoR) allows users to retrieve files without any data loss or corruptions. It uses spot

checking and error correcting codes are used to ensure both "Possession" and "Retrievability". To achieve zero knowledge privacy, researcher [3] proposed Aggregatable Signature Based Broadcast (ASBB). It provides completeness, privacy and soundness. It uses 3 algorithms as Keygen, Gentag and Audit.

3. Conclusion

The researchers carried out a very extensive work. Still there is a scope for future work as a systemthat will be using public key based homomorphic authenticator along with random masking technique to facilitate privacy of user data and enable public auditing of the user's data stored in cloud. To use the public key based homomorphic authenticator through which the TPA can perform the auditing task without asking for the copy of user's data and then integrating it with random masking technique, our protocol promises that TPA will not have necessary information to rebuild the correct group of linear equations and therefore could not have any knowledge about user's data that is stored in the cloud server during the auditing process is having the scope for study.

References

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," in Proc. *ACM CCS*, pp. 598–610 2007.

[2] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing", *in Proc. IEEE INFOCOM*, pp. 525–533. 2010.

[3] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Information Security: *Advances in Cryptology (Asiacrypt)*, vol. 5350, pp. 90-107, Dec. 2008.

[4] B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," University of Toronto, *Tech. Rep*., pp.295-302, 2011.

[5] C. Wang, K. Ren, W. Lou and J. Li, "Towards Publicly Auditable Secure Cloud Data Storage Services," in *IEEE Network Magazine*, vol. 24, no. 4, pp. 19-24, July/Aug. 2010.

[6] T. Prasanthi, C. Balasubramanian, S. Kimsukha Selvi, K. Kala, "An Efficient Auditing Protocol for Secure Data Storage in Cloud Computing" Proceedings of the World Congress on Engineering," *WCE*, vol. 1, pp. 2-4, 2014.

[7] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-Preserving Audit and Extraction of Digital Contents," Cryptology ePrint Archive, Report 2008/186, 2008.

[8] A. Srijanya. K, N. Kasiviswanath, "Data Integrity Verification by Third Party Auditor in Remote Data Cloud," in *International Journal of Soft Computing and Engineering* (IJSCE), vol. 3, no. 5, 2013.

[9] D. Srinivas' "Privacy-Preserving Public Auditing in Cloud Storage Security," in *International Journal of Computer Science and Information Technologies*, vol. 2, no. 6, pp. 2691-2693, 2011.

[10] Murali Krishna, Hemantha Rama, Implementation of TPA for Secured Cloud Storage Data," in *International Journal of Research in Advanced Engineering Technologies*, vol. 2, no. 3, 2014.

[11] Renuka Goyal, Navjot Sidhu," Third Party Auditor: An Integrity Checking Technique for Client Data Security in Cloud Computing," in *International Journal of Computer Science and Information Technologies*, vol. 5, no. 3, pp. 4526-4530, 2014,

[12] T. Paigude and T. A. Chavan, "A survey on Privacy Preserving Public Auditing for Data Storage Security," in *International Journal of Computer Trends and Technology*, vol. 4, no. 3, 2013.