[1]P Raja Sekhar Reddy

[2]K Ravindranath

Enhancing Secure and Reliable Data Transfer through Robust Integrity

Abstract: - Cloud computing has emerged as a highly efficient platform that allows multiple users to access various services through virtualization on a shared physical network. The participants in a Cloud Computing (CC) environment include Cloud Service Providers (CSP), Consumers, Brokers, and Auditors. The advantages of cloud storage, such as universal network access, convenience, and scalability, have led to data owners preferring to store their data on remote servers. However, the transfer of outsourced data has become a critical requirement for cloud users due to the availability of different cloud storage services with varying quality of services.

One major challenge in this context is ensuring the security of secret keys and data integrity. There is no guarantee of data integrity when storing data on an untrusted cloud server. To address this issue, this paper proposes a secure and efficient data integrity verification scheme for cloud storage services. The scheme utilizes a key-homomorphic cryptographic primitive to reduce system complexity and eliminate the need for a public key authentication framework based on a public key infrastructure (PKI) in the data integrity checking protocol. By employing this approach, the proposed method ensures the integrity of remote data stored on cloud servers. Through security analysis and empirical evaluation, it is demonstrated that our scheme is both practical and effective for securely sharing records with multiple owners in cloud computing.

Keywords: Cloud Computing, Cloud Security, public key infrastructure, Data Integrity, Cloud Threads, Cloud Risks, Third party auditor.

I. INTRODUCTION

Data integrity refers to the assurance that digital data is not corrupted and can still be retrieved by authorized users. Data integrity is the preservation of data's stability, accuracy, and reliability. Digital forensics and data assurance are greatly aided by the integrity verification scheme. Because cloud computing is location-independent, the data integrity verification procedure can be a significant task. Many types of research have been done to address the shortcomings of public auditing. This was possible by using data integrity verification methods. Data integrity is a protocol that runs between CSSs(Cloud support service) and users. Private auditing is a data integrity protocol that runs between CSSs and users. Private auditing is where the user must calculate authentication data internally. This scheme requires that the user communicates with CSS to store the data. This can lead to a more computational burden. To communicate with users, a third-party auditor can be launched. Public auditing is where the auditor can either be a TPA or an authentic user. TPAs are used to increase audit effectiveness and decrease computation costs. It is therefore essential to develop a data integrity verification model that can be used for public auditing. Public audibility of outsourced data is possible using two types of data integrity schemes: Evidence of retrievability and Provable data possessions [1]. Ateniese et al. implemented the PDP scheme. The PDP scheme allows clients to send pre-processed data to untrusted servers. It also retains a small amount of meta- data. Later, the client needs that the server show that the data stored has not been changed. If data is corrupted, deleted or altered by malicious users, the PDP scheme can't be properly retrieved. PDP's behaviour is susceptible to financial scratch, data loss and trust loss. PoP is another data integrity method that is similar to PDP. It offers an additional benefit over PDP. The PoP will retrieve corrupted or lost data by using error-correcting codes. Data integrity can be broken down into probabilistic and deterministic groups. The deterministic scheme is need to get the entire file. This scheme is not recommended to large files because it takes longer for integrity verification. The probabilistic system dynamically retrieves data blocks to verify data integrity. This scheme is more suitable for large files, which needs less computation time [2].

AUDITING

An auditor's job is to provide an objective opinion about the facts and evidence that a company controls to meet a particular objective, criteria, requirement or other requirements. An auditor may also provide an opinion on whether

[1] Research Scholar, Koneru Lakshmaiah Education Foundation, Greenfields, Vaddeswaram, A.P., India

[2]Associate Professor ,Dept. of CSE, Koneru Lakshmaiah Education Foundation, Greenfields , Vaddeswaram,A.P.,India

E-mail: [1]prreddy.cvsr@gmail.com, [2] ravindra_ist@kluniversity.in

the control was effective for a specific period. This is also true for cloud compliance audits. Auditors will ask for evidence that controls have been enabled when cloud compliance is required. Auditors will request evidence of controls being enabled. The cloud auditor will be able to give an opinion as to whether controls are in place and, if so, for how long [3].

Auditors use variety methods to gather evidence. These include observation, confirmation, analysis and confirmation. You can combine these procedures to get evidence that will enable auditors form an opinion on the service being audited. These are some examples of tests that were done for each IT area. This is by no means an exhaustive list.

Challenges in cloud computing

An IT security audit is an examination of IT organizations' checks and balances. Auditors evaluate, test and evaluate the organization's systems and practices to determine if they can protect information assets, maintain data integrity, and achieve their business goals. To achieve these goals, IT security auditors need data from both internal and external sources.

Cloud computing can also pose security risks. Cloud infrastructure is the outcome of three-way negotiations between service provider, end users and cloud service providers (CSPs). This allows for productivity to be maintained, while still maintaining some security. CSPs should ensure data security and allow clients access to any Internet service. CSPs must ensure that cloud computing companies meet clients' business objectives, goals, and future requirements [4].

II. PROBLEM STATEMENT

Several public auditing mechanisms have been introduced to check data integrity effectively. During the public audit approach, it fails to maintain identity privacy in common data and findings when significant confidential facts are reported to the auditor general. Once the current machine is accessed, the person is revoked from the system, and the previously signed blocks through that revoked user are revoked with the help of a trusted method. Whereas public auditing requires the current user to first download the previously signed blocks with the help of the cancelled consumer, then validate the blocks, re-sign these blocks, and finally add the new signature to the cloud. This method generates many computing and conversation resources by downloading and verifying blocks, but the block content remains the same. This technique is unsafe because the current one misuses the statistics of the cancelled consumer. The proposed mechanism allows the auditor-general to efficiently control data integrity in the cloud without downloading all the information. This mechanism maintains the confidentiality of shared facts through a proxy resignation mechanism. In this mechanism, blocks previously assigned to the user that were revoked can be re-signed by the current user. The mysterious key can even be provided as a login for security reasons.

III. RELATED WORK

With cloud storage, users will store their data remotely and enjoy the availability of premium packages and offers from a shared pool of configurable computing assets without the burden of storing and maintaining the original knowledge. However, the fact that users do not have physical possession of external information makes securing data integrity in cloud computing a powerful task, especially for users with abnormal computing assets. Also, users should be using cloud storage as if it were on-premises, without any issues with the requirement to verify its integrity.

Zhang et al. [5] A user's characteristics can be dynamically changed in a multi-authority cloud garage system. The user may have the right to acquire some new functions or cancel some existing functions. You have to change your data to access permissions for this reason. However, modern write-attribute strategies rely on agent servers or general performance loss. They are insufficient to solve the problem of managing write-offs to control access to garage device data in the cloud.

Xiaoyu Li et.al [6] In this paper, they proposed to enter fully feature-based console access for multi-cloud storage structure systems with total security. In their proposed scheme, they could all recover the outsourced data if it benefits if there are enough coded attribute keys to gain acceptance in the insurance key and permission concerning the facts of outsourcing. In addition, the suggested chart offers fixed text content for a period of time and a small account price.

Xue et al. [7] Many cryptosystems, including IBE, provide the best management access. It limits users' ability to choose which data is encrypted with a certain degree of security. Encryption functionality based entirely on CP-ABE features is a promising moment designed to handle the right of access to cryptographic data. There are two types of CP-ABE schemes: CP-ABE, where a similar authority administers all jobs. Some government features of CP-ABE come from specific regions and are controlled with the help of specific authorities. CPABE is a multi-authority tool available in the cloud. Customers can apply different alternatives and use record holders to get accurate coverage of various government jobs and percentage information. However, due to the difficulty of abolition, CP-ABE schemes of multiple businesses cannot be applied simultaneously using more than one business to handle the growing popularity of cloud storage device registries. It is recommended to rely on some great undo schemes that rely entirely on a trusted server to undo the ref partition. Remember that the statistics owners cannot trust the cloud servers, so the traditional activity exclusion technology is no longer suitable for cloud storage structures.

Sushmita et.al [8] Handover micro-access processing is a request for data stored on cloud-like servers. Due to the large number of records, decentralized master control plans are preferable to vital plans. Encryption and decoding have a very high rate and are generally ineffective, even when customers access stats for a limited resource. They developed mainly ABE based encryption with fast encryption and external decoding. The basic concept is to divide the encryption into levels, and the initial processing step is offline while the device is not always in use and while the step is online, even if the facts are encoded by the policy.

IV. PROPOSED METHODOLOGY

In this paper, we proposed a secure and effective data integrity verification scheme for cloud storage services and secure data sharing. Research shows that encryption and data integrity are top concerns for most Cloud Users (CU). Researchers have suggested the use of a trusted Third-party auditor (TPA) to address the issues of confidentiality and data integrity. The TPA is able to perform many resource-intensive tasks such as managing encryption keys and checking out data integrity. It is similar to the protocols of RSA DAP, ECC-DAP and the Secret Sharing and Public Verifiable Dynamic Protocol.

DATA INTEGRITY VERIFICATION

Data Integrity Validation permits us to verify the data integrity guaranteed by the data protection procedures. Data Integrity Validation can be utilized to verify the integrity of the data stored in the cloud and shared over the network. A Third-party auditor (TPA) does the data integrity verification. The TPA frequently challenges the server for block-level data validation by sending file names and blocks randomly. On challenge, the root hash code is calculated by the server. Backup processes can be retried and operated as configured. Data Integrity Validation developed at the separate clients before the data transfer will be verified with the Data Integrity Validation developed at the Media Agent after the data transfer is complete, and vice versa. This possibility can be authorized during backup procedures, principal, and mirror servers. The operational follow structure will be given below table.

PROPOSED ARCHITECTURE

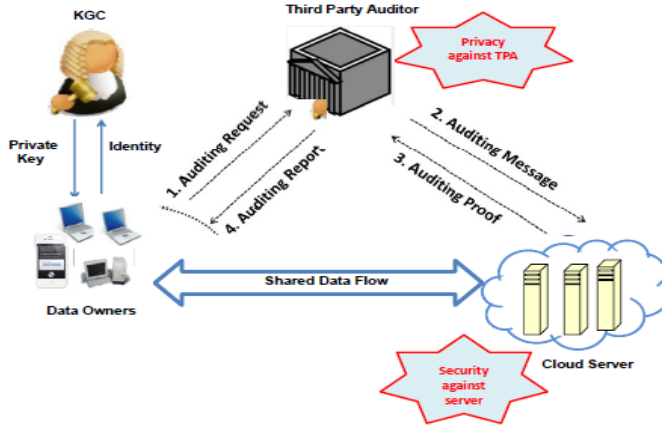

Figure.1 Data integrity Cloud sharing model

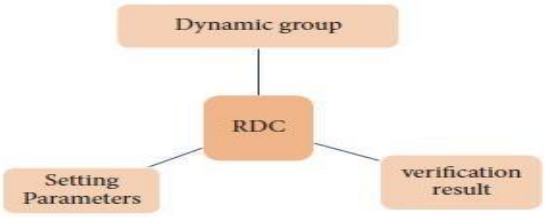

Figure.2 Right Distribution Centre for Cloud Auditing

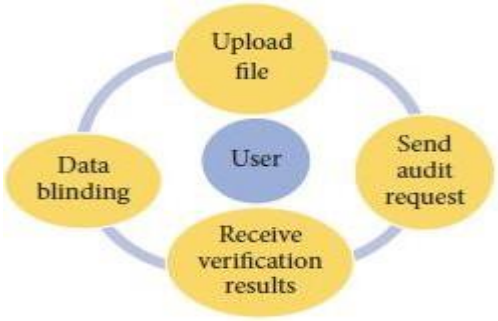

Figure 3 Security in terms of accessing the data using the encrypted keys for Data integrity

Trusted Authority: Trusted authority is a completely dependent component that initializes a device's public key and creates private keys as well as attribute keys for clients.

For example, this could be done through the company director or the Social Security Administration.

CSP: CSP is a quasi-dependent component that provides everyone with a digital region and convenient data storage service with infrastructure in the cloud. The rules entry also adds to the statistics cipher text the co-owners and creates a recoded cipher text for clients.

User: We divide the user's role into the following categories: information owner, information co-owner, statistics distributor, and statistics entry. The stats owner can choose a policy

CONTRIBUTIONS OF THE PROPOSED WORK

Two main entities make up a cloud environment: the CSP is the service provider and the CUs are the service users. These two entities interact using a variety of technologies, including databases, networks and virtualization. The CUs not only use the CSPs' services, but often also outsourcing the sensitive data to the cloud servers. Security issues arise from the different stored in- house.

For CUs the TPA helps reduce the computational burden for CUs but it is still possible that malicious insiders are involved at the TPA. There is a need to find a way to detect malicious activity could result in the sharing/transmission encryption keys between cloud service providers (CSPs) and Cloud users (CU). It is not realistic to assume that all CSPs can be trusted. CSPs can conceal data leakage incidents from CUs in order to keep their good standing. A security incident can be caused by Byzantine failures or server conspiring attacks, as well as malicious data alteration. There are two types of malicious insiders. The first is responsible for debasing the CUs data files stored on individual servers. A malicious insider can modify and read the data on the server.

A CU could become a malicious entity by intentionally violating the terms of the SLA. A CU can sublet services to third-party individuals or organizations once it has received services from a CSP. This can pose serious security risks and raise management questions. Subletting cloud services to third parties can also slow down service delivery from the CSP website. These malicious activities can have a serious impact on the reputation of a CSP and could result in substantial business loss. To detect potential violations of SLA, it is imperative that CUs are periodically audited. A trusted TPA is the best choice for an impartial and fair audit of CUs.

To authenticate all stakeholders, we develop a triangular data PMM. This model aims at ensuring the integrity of the CU's cloud-based data, which can be retrieved anytime. Recent research has mainly focused on the reliability of CSP in terms of security and privacy measures, as well as compliance with SLA. The reliability of the CU or the TPA has not been evaluated. Our model assesses the integrity of the CU in terms of its compliance with the rules set forth by the CSP in their SLA. The TPA also audits the services rendered to the CU, and as such ensures that the integrity of the TPA (i.e., the TPA does not disclose the contents of the CUs from the information obtained during the auditing).

Understanding Security Audit Frameworks for Different Cloud Service Providers

Cloud storage services enable users to store data online and avoid the need for local storage or maintenance. To ensure data in cloud storage is secure, there are many data integrity auditing techniques. To perform data integrity auditing, most, if not all, of the available methods require the user to use his private key. A hardware token, such as a smart card or USB token, is required to activate the private key. A hardware token (e.g., smart card or USB token) is required to activate his private key. It stores the user's private key and password. Many of the current data integrity auditing systems will fail if the hardware token is lost or forgotten. This problem can be solved by a new paradigm in data integrity auditing, which does not require private key storage.

Opportunities and challenges of Cloud Auditors

These evaluations use only current Cloud services and configuration settings as set by service providers. The SOA principle makes cloud provisioning transparent for service users.

Second, it is difficult for researchers to determine impact of Cloud infrastructure resource management on service performance. This is because there is not enough control over the infrastructure configuration. Many service users wish to evaluate the performance of new services to assist them in making their decisions.

Existing works that are focused on predictability and general performance evaluation offer some promising solutions. Research Clouds can be used by researchers to verify validity of performance assessment based on measurement. This allows them to access information about service implementations and control the configuration of service deployments.

ALGORITHMS AND METHOD OF IMPLEMENTATION

This scheme can be used for data integrity auditing. We use a linear sketch to confirm identity. It includes error correction and coding. A new signature scheme was also designed that supports block-less verification and is compatible with the linear sketch. According to security analysis and performance analysis, our proposed scheme is efficient and secure.

Data Encryption

The data owner can outsource their large file to the cloud server to significantly decrease storage load and computational overhead. To control data security from leakage, the user encrypts the data file before transferring the data to the cloud server.

Algorithm 1: **CTEncode**

1. $\boldsymbol{function}\ CTEncode(\boldsymbol{t}, \boldsymbol{T})$
2. $t \leftarrow TEncode(t, T)$
3. $chk \leftarrow false$
4. $\boldsymbol{for}\ i \in [log_2\ T]\ \boldsymbol{do}$
5. $\boldsymbol{if}\ t[i] = 1\ and\ chk = false\ \boldsymbol{then}\ t[i] = 1$
6. $\boldsymbol{else}$
7. $\boldsymbol{chk} \leftarrow true$
8. $t[i] = 0$
9. $\boldsymbol{end\ if}$
10. $\boldsymbol{end\ for}$
11. $\boldsymbol{return}\ t$
12. $\boldsymbol{end\ function}$

Here,

$Time\ \boldsymbol{t}$

$bounded\ system\ life\ time\ T\ are\ input\ and,\ \ bit\ string\ t\ of\ the\ size\ log_2 T$

The specified algorithm is applied to overcome the complexity of the intergalactic. Processing the Cipher text and the Cipher Text Delegation

IMPLEMENTATION

After the data sample has been copied, it can be accessed via all CSP cloud services. The cloud file can then be accessed and credentials provided to verify data integrity. You can either use Python programming or R programming to get performance results

Preliminaries and notation

1. F and F[i] are a file and the ith block of data for F.
2. H (∗): is a hash function.
3. ph (∗): Euler's totient function.
4. m(∗): a Pseudo-Random Function (PRF) which maps: m: {0, 1k x {0, 1}|5. m (∗):

   A pseudo- random function (PRF), which maps: m, 0 x 1k x 10. |5. m (∗): is a pseudo-random function (PRF), that maps: m: 0, 1k, x m: 0, 1, 1. }
5. s (∗) : a Pseudo-Random Permutation (PRP) which maps: s : {0, 1k x {0, 1 … n} − − {0, 1 … n}. |
6. s (*). A pseudo-random permutation (PRP), which maps: S : 0 x 1k x 1… N − − 0… n. |6. s}
7. p and Q: Two different odd prime numbers of equal length.
8. J: Multiplication two prime numbers p or q.
9. r1, r2 : Random numbers taken from the Galois field
10. T and Ti: All block tags and ith tags of T.
11. R is the number of blocks needed for each challenge.
12. kt: Encryption key for tag.
13. kd: Decryption key for tag.
14. Z and Z: Group on integer numbers.
15. c: Number of blocks that are available for challenge operation.
16. r: Number of deleted blocks in total file blocks.

Auditing Framework for Data Division and Reduplication

Auditing's Role

Auditing is essential for organizing, planning, and delivering support. It evaluates, monitors, and assesses third-party providers' and service provider performance (IT Governance 2012). It should use auditing systems to verify confidentiality, data integrity and availability, authentication, reliability, and security. It should be more responsible and contribute to key strategic areas such as customer relations, cost reductions and revenue maximization. Business management includes auditing. Auditing should be about adding value by supporting strategic initiatives and providing valuable insight into the company. Auditing should be actively involved in monitoring, evaluation and improvement of regulatory compliance.

Cloud Auditing

Cloud storage and processing of data does not require a lot of resources and local systems. Cloud storage is an efficient way for users to store their data. Users also prefer cloud storage because they can store as many data as they want without restriction. The cloud transfers the application software and databases into large, centralized data centers. However, this can make it less trustworthy because of security concerns like old IP addresses, etc.

Auditing is the process of evaluating and collecting evidence to determine if a computer system's safety, efficiency, security, and data integrity. To verify and maintain data integrity, an auditor must audit the data stored on the cloud. There are many ways to ensure data security.

PPPA for Secure Cloud Storage

Fog empowers to own right to use up to several services. Costumer's loo get entry to sensational supplies on-demand along with getting pleasure from purposes along with providers. Web mustiness sees journal integrity in addition to the certificate. This can be a shocking key outcome. As mentioned above, the effect will be handled with semi-public accountancy tubercle for which concerns mediator examiners (TPA). This technique promotes record kinetics in addition to real- world scrutinize skills. Management consulting is employed to trace track record differences, false positives, furthermore insertions. The auditing procedure is supported by the system, which includes data dynamics, public audibility, and multiple TPA. HARS is used to create ring signatures. Merkle Hash Trees are used to improve block-level authentication. Through the Batch auditing process, the TPA can perform audits simultaneously for multiple users.

Dynamic data operations

The important dynamic data operations are divided into Data Insertion (DI), Data Deletion (DD) and Data Modification (DM).

Data Insertion (DI)

Data insertion operation inserts a new block after the specified position of a file F. Mostly it does not alter logic structure of customers data. Suppose the client wants to insert a new block, this protocol supports insertion of a new block b_{new} after a given block b_i into a file F. As a first step, user initially generates new data information (VER_{new}, TS_{new}) for new block b_{new} contains new version and timestamp of a new file block to be inserted. Then client sends the update request to $update_{ins}$ (F, DI, i, VER_{new}, TS_{new}) to Trusted Third Party Auditor (TTPA)

Algorithm Steps:

1: begin

2: Update record in ITS

3: begin

4: user generates new data information

(VER*new*, TS*new*) for the new

block *bnew*.

5: sends update request update*ins*(*F*, DI, *i*, VER*new*, TS*new*) TTPA

6: end

7: TTPA performs the following after

receiving update*ins* request

8: begin

9: TTPA finds the last record in ITS and inserts the new one after it

10: update both VER*new*, TS*new*, increment pointer by 1

11: end
12: update stored data in cloud

13: begin

14: user generates new signature Tnew for the new block b**new**

15: sends update request updateins(F, DI, i, bnew (VERnew, TSnew), Tnew->CSP

16: CSP generates new version of file Fnew and tag set Tnew = Fnew||n||sigSsk(Fnew||n)

17: end

18: end

Data Deletion (DD)

Data deletion operation refers to deletion of a specified block and requires moving all the blocks after deletion. Suppose the client wants to delete a block, this protocol supports deletion of a particular block bi from the file F. As a first step, user initially generates request for deletion as update del (F, DD, i) to Trusted Third Party Auditor (TTPA). The request consists of a file; DD refers to data deletion and a block number to be deleted. TTPA carries out the update request by deleting a specified block upon receiving the request from clients.

Algorithm

1: begin

2:update record in ITS

3: begin

4: user sends update request update del (F, DD, i) -> TTPA

5: TTPA deletes ith record in ITS

6: decrement pointer by 1

7: end

8: update stored data in cloud

9: begin

10: user sends update del (F, DD, i) o CSP

11: CSP gets new file version Fnew and tag set 7new =Fnew||n||sigSsk(Fnew||n) 12: end

Data Modification (DM)

In cloud data storage data modification is the most important one among the three operations. This protocol supports the replacing of a specified block with a new block. This is the most frequently used operation in cloud storage. Assume that the client wants to modify the i^{th} file block b_i in to b_{newc}. Data modification operation replaces the specified block of bi to b_{new}^{c}. As a first step, client generates the new information (VER_{new}^{c}, TS_{new}^{c}) for the particular new block b_{new}^{c}. Then it sends the update request $update_{mod}$(F, DM, i, VER_{new}^{c}, TS_{new}^{c}) to the auditor (TTPA).

The server performs the update operation on their side upon receiving the request from user

i) replaces the old file with the new file version F_{new}^{c}

Generates tag set F_{new}^{c} for the new block b_{new}^{c} as F_{new}^{c} ||n||$sig_{Ssk}(F_{new}^{c}$|| n).

Then server sends the update request modification proof to the client

Algorithm

1: begin

2: user generates new (VER_{new}^{c}, TS_{new}^{c}) information for the block b_{new}^{c}

3: update record in ITS

4: begin

5: Sends update request updatemod(F, DM, i, VER_{new}^{c}, TS_{new}^{c}) for b_{new}cĺ TTPA

6: end

7: TTPA performs the following after receiving $update_{mod}$ request

8: begin

9:TTPA finds the appropriate record in ITS 10: replace (VER, TS) with VER_{new}^{c}, TS_{new}^{c} 11: end

12: update stored data in cloud

13: begin

14: user generates new signature V_{new}^{c} for the new block b_{new}^{c}

15: sends update modification request updatemod(F, DM, i,b_{new}(VER_{new}, TS_{new}), b_{new} (VER_{new}^{c}, TS_{new}^{c}), V_{new}^{c} o CSP

16: CSP generates new version of file F_{new}^{c} and tag set TS_{new}^{c} = F_{new}^{c} ||n||$sig_{Ssk}(F_{new}^{c}$|| n) after replacing the old block with a new one

17: end

18: end

V. RESULTS AND ANALYSIS

We rather examine the safety of our proposed method. Later, we provide the theoretical, computational complexity investigation and approximate efficiency comparison through numeric investigation. In this experiment, we execute our suggested performance evaluation method and compare it with existing cryptographic primitives. Especially all of the simulation investigations are taken out on a Desktop equipped with the Windows operating system with i5 or i7 processors with 8 GB RAM. Also, we execute the corresponding cryptographic algorithms with the available, safe sockets layer library and the pairing-based cryptography approaches. Also, we ignore the transmission overhead and only calculate the time overhead of performing the main processes in each phase.

Data Encryption Computational Overhead

The major measures are data encryption, and the key generation process is closely associated with the encrypted file size. Thus, in this experimentation, we improve the encrypted file size from 1MB to 10MB with a set of 1 MB. Also, we assume that the total number of data blocks is 20,000. Later, we estimate the approximate time overhead in milliseconds, which is illustrated in Figure 4.

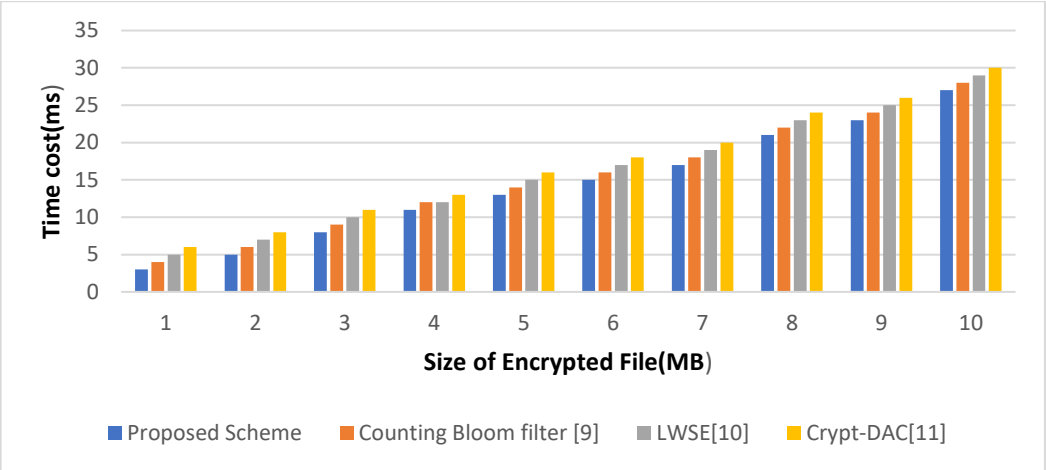

Figure 4 Time cost of data encryption

As shown in Figure 4, we can readily find that the overhead approximatively linearly increases with the encrypted file size. In this graph, both schemes' growth rates of time cost are almost identical. However, the computational overhead of our proposed scheme is less than that of the existing scheme [9].

Computational Overhead of Data Outsourcing

The primary investigations are hash computations, auditing measures and signature generation. To better demonstrate the data storage comparison in this section, we would improve the number of cloud data blocks from 200 to 2,000 with a stage for 200 blocks. Later, we estimate the time cost in milli seconds(ms), shown in Figure 5.

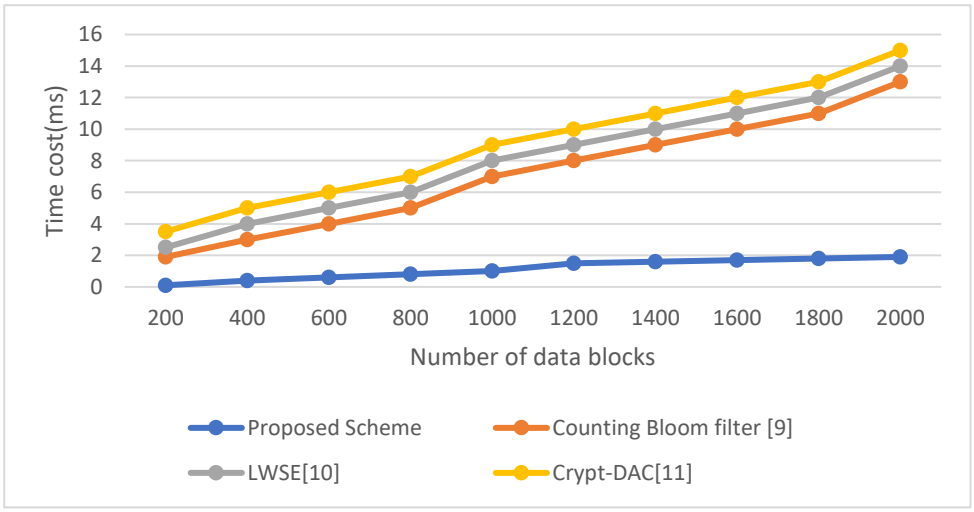

Figure 5 Time cost of data storage

As shown in the figure 5. Firstly, we can detect that the overhead of our proposed method is less than that of the existing method.

Computational Overhead of Data Transfer

In this investigation, the main time overhead is from the estimates of demonstrating the data integrity checking to the number of transmitted cloud data blocks. Due to the fluctuation of the network, we disregard the communication overhead. Thus, we increase the number of transmitted cloud data blocks from 10 to 100 with a step for 10. Later, we estimate the time cost of achieving the major operations, illustrated in Figure 6.

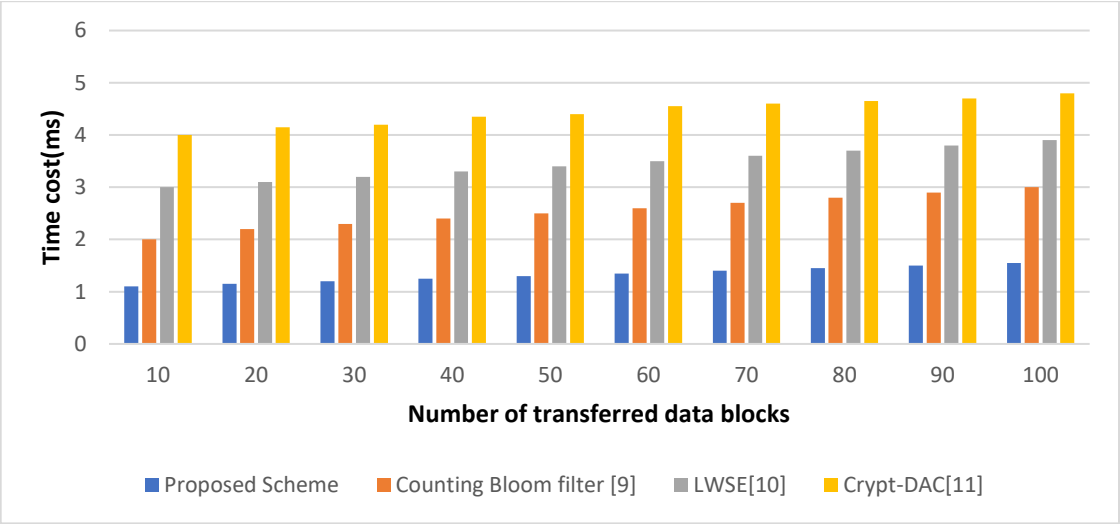

Figure 6 Time cost of data transfer

Also, we can efficiently assume that the total time overhead of our suggested method is much less than that of the existing method [9]. With the experimental results, the presented method has more advantages in the efficiency of every stage and more advantages in total efficiency.

VI. CONCLUSION

In this paper, we proposed a secure and effective data integrity verification method for cloud storage services by supplying key-homomorphic cryptographic primitive to reduce the system complexity. Our proposed approach enables the user to share the data from one cloud server to another and then examine the transmitted cloud data integrity on the target cloud server. This proposed protocol can provide data security in the case of data integrity auditing. In addition, the experimental results shows that the proposed scheme has strong advantages in the time overhead of proof generation compared with the available Bloom filter-based technique. With the advancement of cloud storage, the data owner may wish to transfer outsourced data from one cloud to two or more target clouds at the same time.

REFERENCES

[1] Y. Wang and J. Han, 2017, “Full verifiability for outsourced decryption in attribute-based encryption", IEEE.

[2] X. Tao and C. Yang, 2018, “New publicly verifiable cloud data deletion scheme with efficient tracking,”, pp. 359-372.

[3] K., Wang C, et al, 2016, “Enabling cloud storage auditing with verifiable outsourcing of key updates”. IEEE, pp.1362–1375.

[4] Qin, J., Yu, J., et al, 2019, “Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage”. IEEE, pp.331–346.

[5] Q. Zhang, L. T. Yang, and Z. Chen, “Privacy preserving deep computation model on cloud for big data feature learning,” IEEE Transactions on Computers, vol. 65, no. 5, pp. 1351-1362, 2016.

[6] Xiaoyu Li, Jie Chen, 2017, “Two-Factor Data Access Control with Efficient Revocation for Multi-Authority Cloud Storage Systems", pp. 393-405

[7] K. Xue, J. Hong, “Combining data owner-side and cloud-side access control for encrypted cloud storage,” IEEE Transactions on Information Forensics and Security, vol. 13, no. 8, pp. 2062–2074, 2018.

[8] Sushmita Ruj, 2015, "Decentralized Access Control on Data in the Cloud with Fast Encryption and Outsourced Decryption", pp. 1-6.

[9] C. Yang and Y. Wang, 2020, “Secure data transfer and deletion from counting Bloom filter in cloud computing,” Chinese Journal of Electronics, vol. 29, no. 2, pp. 273–280, 2020.

[10] J. Chang and Bian G, 2020, “Certificateless provable data possession protocol for the multiple copies and clouds case,”, pp. 102958–102970.