iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** XI **Month of publication:** November 2021

DOI: https://doi.org/10.22214/ijraset.2021.38834

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XI Nov 2021- Available at www.ijraset.com

Third Party Public Auditing Scheme for Cloud Storage

Mr.Vaishnav P. Surwase[1], Prof. A.H Rokade[2]
[1]*ME Student, Department of Computer Science & Engineering, BAMU University, SYCET, Aurangabad*
[2]*Head of Department, Department of Computer Science & Engineering, BAMU University, SYCET, Aurangabad*

Abstract: ***Thus the new auditing scheme has been developed by considering all these requirements. It consist of three entities: data owner, TPA and cloud server. The data owner performs various operations such as splitting the file to blocks, encrypting them, generating a hash value for each, concatenating it and generating a signature on it. The TPA performs the main role of data integrity check. It performs activities like generating hash value for encrypted blocks received from cloud server, concatenating them and generates signature on it. It later compares both the signatures to verify whether the data stored on cloud is tampered or not. It verifies the integrity of data on demand of the users. The cloud server is used only to save the encrypted blocks of data. This proposed auditing scheme make use of AES algorithm for encryption, SHA-2 for integrity check and RSA signature for digital signature calculation. In this philosophy, users of cloud storage services no longer physically maintain direct control over their data, which makes data security one of the major concerns of using cloud. Existing research work already allows data integrity to be verified without possession of the actual data file. When the verification is done by a trusted third party, this verification process is also called data auditing, and this third party is called an auditor. As a result, every small update will cause re-computation and updating of the authenticator for an entire file block, which in turn causes higher storage and communication overheads. In this paper, we provide a formal analysis for possible types of fine-grained data updates and propose a scheme that can fully support authorized auditing and fine-grained update requests. Basedon our scheme, we also propose an enhancement that can dramatically reduce communication overheads for verifying small updates***
Keywords: Cloud computing, big data, data security, authorized auditing, fine-grained dynamic data update

I. INTRODUCTION

As the IT technologies are growing day by day, the need of computing and storage resources are rapidly increasing. To invest more and more equipments is not an economic method for an organization to satisfy the even growing computational and storage need. So Cloud Computing has become a widely accepted paradigm for high performance computing, because in Cloud Computing all type of IT facilities are provided to the users as a service. Cloud computing is a category of sophisticated on-demand computing services initially offered by commercial providers, such as Amazon, Google, and Microsoft.

In Cloud Computing the term Cloud is used for the service provider, which holds all types of resources for storage, computing etc. Mainly three types of services are provided by the cloud. First is Infrastructure as a Service (IaaS), which provides cloud users the infrastructure for various purposes like the storage system and computation resources. Second is Platform as a Service (PaaS), which provides the platform to the clients so that they can make their applications on this platform. Third is Software as a Service (SaaS), which provides the software to the users; so users don't need to install the software on their own machines and they can use the software directly from the cloud. The foremost issues in cloud data security include data privacy, data protection, data availability, data location, and secure transmission. Threats, data loss, service disruption, outside malicious attacks, and multi tenancy issues are the security challenges included in the cloud. Data integrity in the cloud system means preserving the integrity of stored information. The data should not be lost or modified by unauthorized users. Cloud computing providers are trusted to maintain data integrity and accuracy of data. Data confidentiality is also important aspect from user's point of view because they store their private or confidential data in the cloud. Authentication and access control strategies are used to ensure data confidentiality. The data confidentiality could be addressed by increasing the cloud reliability and trustworthiness in Cloud computing. Therefore security, integrity, privacy and confidentiality of the stored data on the cloud should be considered and are important requirements from user's point of view [4]. To achieve all of these requirements, new methods or techniques should be developed and implemented. The three network entities viz. the client, cloud server and TPA are present in the cloud environment. The client stores data on the storage server provided by the cloud service provider (CSP). TPA keeps a check on client's data by periodically verifying integrity of data on-demand and notifies client if any variation or fault is found in client's data. Figure 1 shows the cloud data storage architecture.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XI Nov 2021- Available at www.ijraset.com

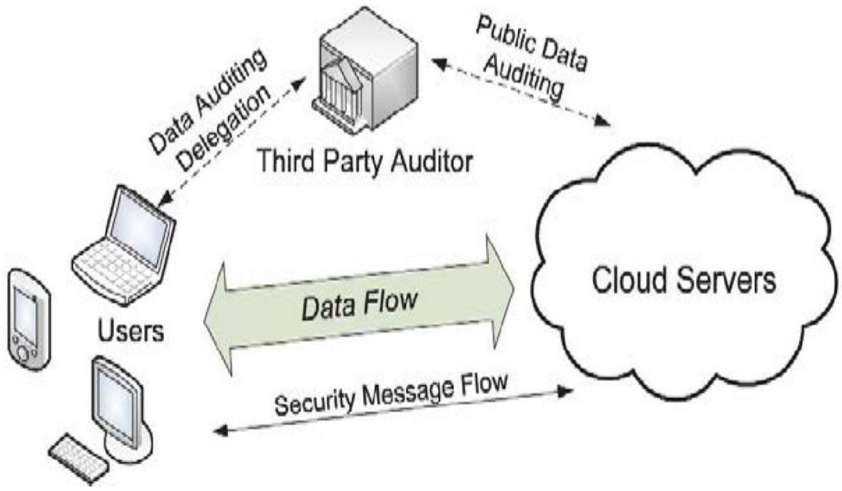


Fig. 1. Cloud Data Storage Architecture

The rest of the paper is organized as follows. Section 2 introduces Literature survey. The Proposed system is described in Section 3 whereas Section 4 provides the concluding remarks and future work.

II. LITERATURE SURVEY

Although existing data auditing schemes already have various properties (see Section 2), potential risks and inefficiency such as security risks in unauthorized auditing requests and inefficiency in processing small updates still exist. In this paper, we will focus on better support for small dynamic updates, which benefits the scalability and efficiency of a cloud storage server. To achieve this, our scheme utilizes a flexible data segmentation strategy and a ranked Merkle hash tree (RMHT). Meanwhile, we will address a potential security problem in supporting public verifiability to make the scheme more secure and robust, which is achieved by adding an additional authorization process among the three participating parties of client, CSS and a third-party auditor (TPA).

Research contributions of this paper can be summarized as follows:

1) For the first time, we formally analyze different types of fine-grained dynamic data update requests on variable-sized file blocks in a single dataset. To the best of our knowledge, we are the first to propose a public auditing scheme based on BLS signature and Merkle hash tree (MHT) that can support fine-grained update requests. Compared to existing schemes, our scheme supports updates with a size that is not restricted by the size of file blocks, thereby offers extra flexibility and scalability compared to existing schemes.
2) For better security, our scheme incorporates an additional authorization process with the aim of eliminating threats of unauthorized audit challenges from malicious or pretended third-party auditors, which we term as ‘authorized auditing’. We investigate how to improve the efficiency in verifying frequent small updates which exist in many popular cloud and big data contexts such as social media. Accordingly, we propose a further enhancement for our scheme to make it more suitable for this situation than existing schemes. Compared to existing schemes, both theoretical analysis and experimental results demonstrate that our modified scheme can significantly lower communication overheads

In this table, comparison is done on various factors such as method used, whether supports public auditing, privacy preserving, data dynamic and batch auditing. It also shows if the integrity and confidentiality of data stored in the cloud server is maintained or not. From the table, it is clearly seen that different methods have been implemented to check the integrity of the data. But each method has some issues connected with it. The existing methods succeeded in providing privacy preserving along with public auditing but failed to maintain the confidentiality of data. Therefore it is necessary to develop an efficient and secure auditing scheme which can perform public auditing effectively by maintaining both the integrity and confidentiality of data.

Table 1. Comparison of Existing Privacy Preserving Public Auditing schemes.

Research papers	Method used	Supports Public Auditing	Supports Privacy Preserving	Supports Data dynamics	Supports Batch auditing	Maintaining Integrity of data	Maintaining Confidentiality of data
Privacy Preserving Public auditing for data storage security in Cloud computing[9]	HLA with random masking	Yes	Yes	No	No	Yes	No
Privacy Preserving public Auditing for Secure Cloud Storage[6]	HLA with BLS signature	Yes	Yes	No	Yes	Yes	No
Privacy Preserving Public Auditing for Secure Cloud Storage[7]	HLA with BLS signature	Yes	Yes	No	Yes	Yes	No
Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing[10]	HLA with BLS signature along with MHT	Yes	Yes	Yes	Yes	Yes	No
Towards Secure and Dependable Storage Services in Cloud Computing[8]	Homomorphic tokens + Erasure code	Yes	Yes	Yes	No	Yes	No
Secure and efficient privacy preserving public auditing scheme for cloud storage[11]	HLA with BLS signature	Yes	Yes	No	Yes	Yes	No
Cloud Server Storage using TPA[2]	Merkle Hash tree	Yes	Yes	Yes	No	Yes	No
Privacy preserving & Public auditing service for Data storage in Cloud Computing[5]	MHT+RSA algorithm	Yes	Yes	No	No	Yes	Yes
Privacy Preserving & Batch auditing in Secure Cloud data storage using AES[3]	HLA with random masking+AES algorithm	Yes	Yes	No	Yes	Yes	Yes
Privacy preserving Public auditing in cloud using HMAC algorithm[1]	HMAC algorithm	Yes	Yes	No	No	Yes	No

III. PROPOSED SYSTEM

Integrity verification for outsourced data storage has attracted extensive research interest. The concept of proofs of retrievability (POR) and its first model was proposed by Jules . Unfortunately, their scheme can only be applied to static data storage such as archive or library. In the same year, Ateniese, et al. proposed a similar model named 'provable data possession' (PDP) . Their schemes offer 'blockless verification' which means the verifier can verify the integrity of a proportion of the outsourced file through verifying a combination of pre-computed file tags which they call homomorphic verifiable tags (HVTs) or homomorphic linear authenticators (HLAs). Work by Shacham, provided an improved POR model with stateless verification. They also proposed a MAC-based private verification scheme and the first public verification scheme in the literature that based on BLS signature scheme.

In their second scheme, the generation and verification of integrity proofs are similar to signing and verification of BLS signatures. When wielding the same security strength (say, 80-bit security), a BLS signature (160 bit) is much shorter than an RSA signature (1024 bit), which is a desired benefit for a POR scheme. They also proved the security of both their schemes and the PDP scheme by Ateniese, et al. [9], [10]. From then on, the concepts of PDP and POR were in fact unified under this new compact POR model. Ateniese, et al. extended their scheme for enhanced scalability [8], but only partial data dynamics and a predefined number of challenges is supported. In 2009, Erway, et al. proposed the first PDP scheme based on skip list that can support full dynamic data updates. However, public auditability and variable-sized file blocks are not supported by default. Wang, et al. [6] proposed a scheme based on BLS signature that can support public auditing (especially from a thirdparty auditor, TPA) and full data dynamics, which is one of the latest works on public data auditing with dynamics support. However, their scheme lacks support for finegrained update and authorized auditing which are the main focuses of our work. Latest work by Wang added a random masking technology on top of [6] to ensure the TPA cannot infer the raw data file from a series of integrity

The proposed scheme consists of three basic entities; they are data owner, cloud server storage and TPA. The data owner or the user is responsible for splitting the file into blocks, encrypting those using AES algorithm, generating a SHA-2 hash value for each, concatenating the hashes and generates a RSA signature on it. The cloud server is used to store the encrypted blocks of files. When the client or data owner request for data auditing to the TPA, it immediately request for the encrypted data from the cloud server. After receiving the data, it generated the hash value for each block of encrypted files. It uses the same SHA-2 algorithm which was used by client. It later concatenate those hash values and generates a RSA signature for that file. In the Verification process, the signature generated by TPA and the one stored in the TPA which is provided by the data user are compared by the TPA. If they matches with each other it means that the data is intact and data is not been tampered by any outsider or attacker. If it does not matches then it indicates that the data integrity has been affected or tampered. The result for the data integrity check is provided to the data owner. Figure 2. shows the Architecture for the Proposed Auditing scheme.

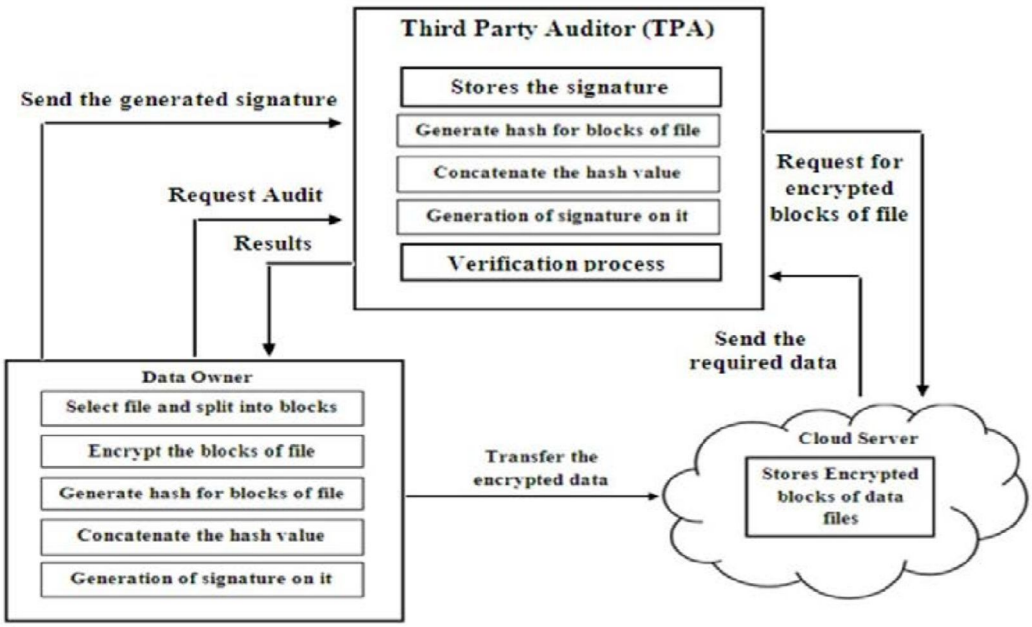


Fig. 2. Proposed Auditing scheme Architecture

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XI Nov 2021- Available at www.ijraset.com

Data owner is an important part of our proposed system. It performs most of the responsibility related to the data.
In the proposed auditing scheme, the data owner first performs login and registration with cloud server and TPA. The new user has to firstly register itself by filling the registration form and be the active member of the system. A message for successful registration will be provided. If a user is already the member of the system then he or she can perform login process. If the user name and password exist in the database, then they will be login successfully for being valid users or else they will receive an error message.

A. Our Scheme is Described in three Parts

1) Setup: The client will generate keying materials via KeyGen and FileProc, then upload the data to CSS. Different from previous schemes, the client will store a RMHT instead of a MHT as metadata. Moreover, the client will authorize the TPA by sharing a value sig_{AUTH}.

2) Verifiable Data Updating: The CSS performs the client's fine-grained update requests via PerformUpdate, then the client runs VerifyUpdate to check whether CSS has performed the updates on both the data blocks and their corresponding authenticators (used for auditing) honestly.

3) Challenge, Proof Generation and Verification: Describes how the integrity of the data stored on CSS is verified by TPA via Gen Challenge, Gen Proof and Verify.

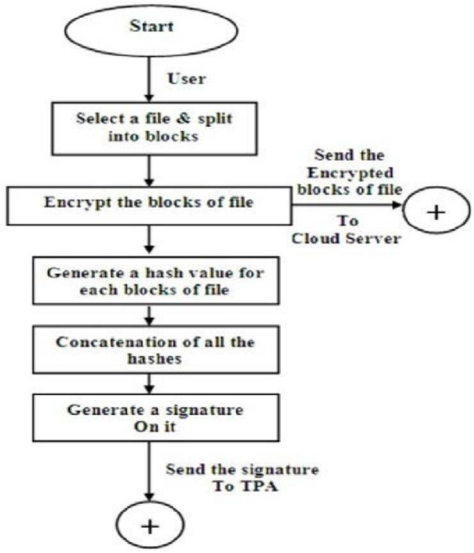


Fig. 3. Flowchart for working of Data Owner

The data owner make use of cloud storage to store the encrypted form of data. As the data is stored in encrypted form, so the cloud server has zero knowledge about the data. As well as if the cloud server turns into malicious server or is attacked by any outside attacker, the data will not be retrieved easily as it is in the encrypted form and it is not aware about the encryption algorithm implemented by the data owner. In the proposed scheme, to perform the task of data auditing a TPA is been used for this purpose. TPA performs data auditing either periodically or on demand by the client. On receiving the auditing request from user or data owner, the TPA starts its auditing process. TPA also stores the signature which has been generated by data owner. The TPA follows the same process performed by data owner such as generating hash for encrypted blocks of data files, concatenating them and generating signature on it. Later it compares the two signature in verification process. If it matches then it means the integrity of data is maintained and otherwise not maintained. This means that data is not been tampered or changed. The results for the same is provided to the data owner by the TPA. The following Figure 4. shows the working of the TPA in our proposed auditing scheme.
In the challenge/verification process of our scheme, we try to secure the scheme against a malicious CSS who tries to cheat the verifier TPA about the integrity status of the client's data, which is the same as previous work on both PDP and POR. In this step, aside from the new authorization process (which will be discussed in detail later in this section), the only difference compared to [6] is the RMHT and variable-sectored blocks. Therefore, the security of this phase can be proven through a process highly similar with [6], using the same framework, adversarial model and interactive games defined in [6]. A detailed security proof for this phase is therefore omitted here

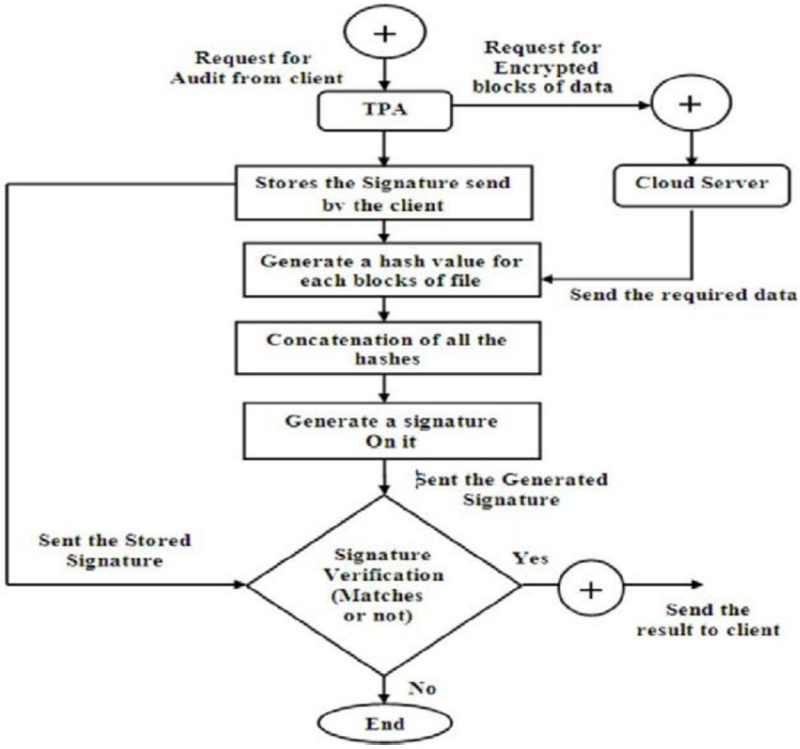


Fig. 4. Flowchart for working of TPA

We investigated the performance improvement of the modification introduced in Section 4.5. We used 3 pieces of random data with sizes of 100 bytes, 140 bytes and 180 bytes, respectively, to update several blocks that contain 10 to 50 standard 20-byte sectors each. Data retrieval is a key factor of communication overheads in the verifiable update phase. For each update, we recorded the total amount of data retrieval for both our modified scheme and our basic scheme. The results in comparison are shown in Fig. 5. We can see that our modified scheme always has better efficiency with respect to data-retrieval-invoked communication overheads, and the advantage is more significant for larger updates. However, for an update of the same size, the advantage will decrease with the increase of js_ij where a larger number of sectors in the original file are needed to be retrieved. Therefore, the block size needs to be kept low if less communication in verifiable updates is highly demanded.

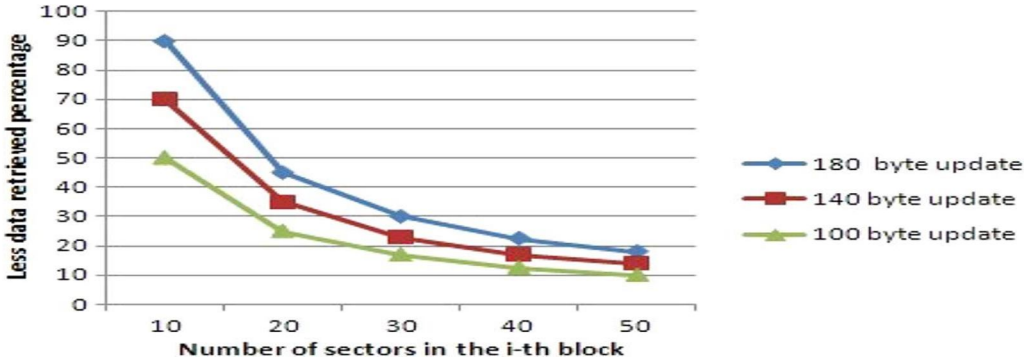


Fig: 5 . Percentage in saving of communication overhead in data retrieval in the modified scheme, compared to our basic scheme

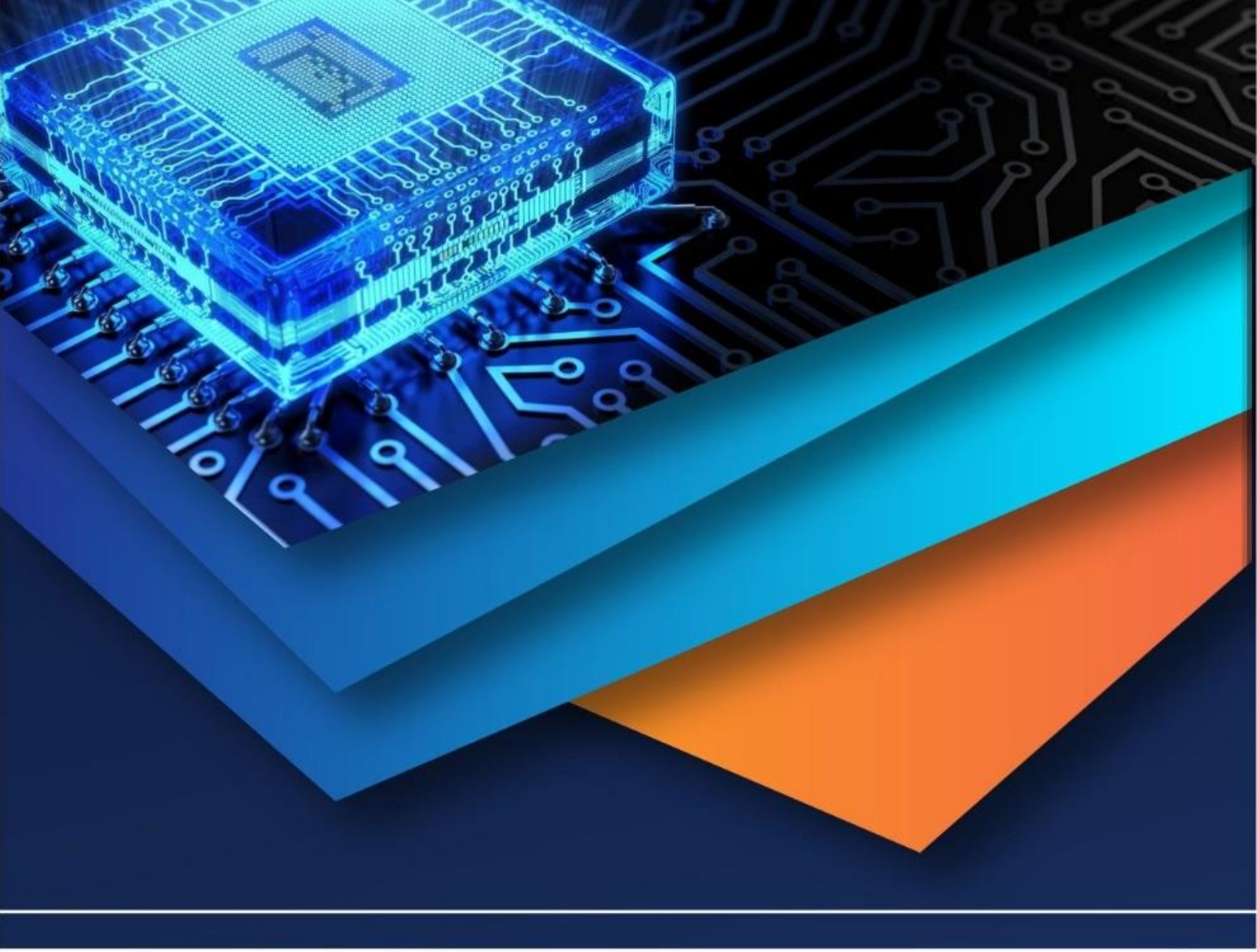
From the experimental results on small updates, we can see that our scheme can incur significantly lower storage overhead while our modified scheme can dramatically reduce communication overheads compared to the existing scheme. In practice, the important parameter s_{max} should be carefully chosen according to different data size and different efficiency demands in storage or communications. For example, for general applications with a similar scale (1GB per dataset and frequent 140-byte updates), a choice of s_{max} ¼ 30 will allow the scheme to incur significantly lowered overheads in both storage and communications during updates.

IV. CONCLUSION

A secure and efficient privacy preserving public auditing scheme is been proposed. It achieves privacy preserving and public auditing for cloud by using a TPA (Third Party Auditor), which does the auditing without retrieving the data copy, hence privacy is preserved. The data is split into parts and then stored in the encrypted format in the cloud storage, thus maintaining the confidentiality of data. The data integrity is verified by TPA on request of the client by verifying both the signatures. It only checks whether the stored data is tampered or not and informs about it to the user. An attempt is made to overcome the limitations of the existing auditing scheme. All the modules in the system are implemented to develop an effective auditing scheme. In future, we would like to perform data dynamic operations such as updation , deletion and insertion of data.

REFERENCES

[1] S Ezhil Arasu, B Gowri, and S Ananthi. Privacy-Preserving Public Auditing in cloud using HMAC Algorithm. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277, 3878, 2013.
[2] IK Meenakshi and Sudha George. Cloud Server Storage Security using TPA. International Journal of Advanced Research in Computer Science & Technology (IJARCST) ISSN: 2347-9817, 2014.
[3] Jadhav Santosh and B.R nandwalkar. Privacy Preserving and Batch auditing in Secure Cloud Data Storage using AES. Proceedings of 13th IRF International Conference, ISBN: 978-93-84209-37-72014.
[4] Zissis, Dimitrios, and Dimitrios Lekkas. Addressing cloud computing security issues. Future Generation computer systems 28.3 (2012): 583-592.
[5] Tejaswini, K. Sunitha, and S. K. Prashanth. Privacy Preserving and Public Auditing Service for Data Storage in Cloud Computing. Indian Journal of Research PARIPEX, 2(2), 2013.
[6] Cong Wang, Sherman SM Chow, Qian Wang, Kui Ren, and Wenjing Lou. Privacy Preserving Public Auditing for Secure Cloud Storage. http://eprint.iacr.org/2009/579.pdf
[7] Cong Wang, Sherman SM Chow, Qian Wang, Kui Ren, and Wenjing Lou. Privacy Preserving Public Auditing for Secure Cloud Storage. Computers, IEEE Transactions on, 62(2):362–375, 2013.
[8] Cong Wang, Qian Wang, Kui Ren, Ning Cao, and Wenjing Lou. Toward secure and dependable storage services in cloud computing. Services Computing, IEEE Transactions on, 5(2):220–232, 2012.
[9] Cong Wang, Qian Wang, Kui Ren, and Wenjing Lou. Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing. In INFOCOM, 2010 Proceedings IEEE, pages 1–9. IEEE, 2010.
[10] Qian Wang, Cong Wang, Kui Ren, Wenjing Lou, and Jin Li. Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing. Parallel and Distributed Systems, IEEE Transactions on, 22(5):847–859, 2011.
[11] Solomon GuadieWorku, Chunxiang Xu, Jining Zhao, and Xiaohu He. Secure and efficient privacy-preserving public auditing scheme for cloud storage. Computers & Electrical Engineering, 40(5):1703–1713, 2014.
[12] Mell, Peter, and Tim Grance. The NIST definition of cloud computing. (2011).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY