e-ISSN: 2349-9745
p-ISSN: 2393-8161

Scientific Journal Impact Factor (SJIF): 1.711

International Journal of Modern Trends in Engineering and Research

www.ijmter.com

Trust Enhancing Approach for Cloud Computing

Palvinder Singh[1] ,Er. Anurag Jain[2]
[1]Galaxy Institute of Technology & management, Karnal
[2]Assist. Prof.Geeta Institute of Management & technology, Kurukshetra

Abstract: - Cloud computing is a web based computing that allows sharing of services. Storing our information in cloud might not be absolutely trustworthy. Since consumer does not have copy of all hold on information, he has to depend upon Cloud Service Provider (CSP). This paper proposes a versatile distributed storage integrity auditing mechanism, utilizing the Homomorphic token that assures a trustworthy and secured cloud storage service. To ensure the correctness of data, we tend to take into account the task of permitting a Third Party Auditor (TPA), on behalf of the cloud client to verify the integrity of the data stored in the cloud. The identification of corrupted server is done instantly, which helps in easy retrieval of data. The proposed work allows cloud clients to audit the cloud storage with very lightweight communication and computation cost. The proposed work assures secured communication between server and client using strong authentication. Proposed solution also provides the confidentially and integrity of transmitted data between users and cloud service providers. The proposed work also provides Privacy Preserving Multi-Keyword Search over Encrypted Cloud Data. An authorized user can also search multi keywords over encrypted data by preserving his privacy.

Keywords: - Batch Auditing, Data Integrity, Data Dynamics, Seb´e'et al's protocol, Public Verifiability, Privacy, Third-Party Auditor, Multi Keyword Search

1. INTRODUCTION

With the advent of cloud computing as a new computing paradigm, more flexible services are transparently provided to worldwide users. The cloud is a dynamic environment where multiple systems interact. The basic idea in the cloud computing is to move computing tasks from individual systems into the cloud, which provides hardware and software resources over the Internet. By tapping into cloud infrastructure, users can gain fast access to best-of-breed applications and drastically boost computing resources in a cost-effective way. Institutions can improve their information technology's agility and reliability, and obtain device and location independence. Computing resources such as servers, computer networks, databases for storage, and software applications can be configured for the customer needs and are available as shared resources in the cloud. The customers of the cloud get on demand network access to these resources and can be easily provisioned with minimum interaction of the service provider [2]. A main advantage of cloud computing is that the customers can avoid capital expenditure on hardware, software, and services but pay for only what they use to a cloud provider.

However, we need to closely consider the security issues concerning the processed information since the traditional system boundaries are not present in the cloud. The privacy and integrity of the information being processed over the cloud is generally at risk due to the multiple parties accessing this information. When sending our data to be processed in the cloud, we give control of the data to a remote party that might not necessarily address our security concerns. In particular, it is crucial to ensure the privacy and integrity of the information while processing the information over the cloud where previously unknown parties (e.g., remote systems and system administrators) may be present [1].

Storing data in the cloud gives rise to the issue of data integrity verification at entrusted servers. If a client can log in from any location to access data and applications, it is possible that client's privacy could be compromised. Cloud computing companies will need to find ways to protect client privacy. To ensure client's privacy, data file handling mechanism is audited by a secure third-party which was previously discussed as storage service provider [3]. The persona of Third Party Auditor (TPA) is listed as follows:

- Reduce data owner's burden in managing the data.
- Ensure the client that the data stored in the cloud is intact and data integrity is maintained.
- Aid in achieving high economies of scale through customer satisfaction [4].

In this paper, the main contributions of the proposed Seb´e et al.'s protocol are:

(1) A remote data integrity checking protocol for cloud storage, which can be viewed as an adaptation of Seb´e et al.'s protocol [6]. The proposed protocol inherits the support of data dynamics from and supports public verifiability and privacy against third party verifiers, while at the same time it doesn't need to use a third-party auditor.
(2) A security analysis of the proposed protocol, which shows that it is secure against the untrusted server and private against third party verifiers, is given [6].
(3) Compared to many of its predecessors, which only provide binary results about the storage status across the distributed servers, the proposed scheme achieves the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving server(s) [7].
(4) Unlike most prior works for ensuring remote data integrity, the new scheme further supports secure and efficient dynamic operations on data blocks, including: update, delete and append [6].

2. PROBLEM STATEMENT

2.1 System Model

Representative Network architecture for cloud data storage is illustrated in Fig. 1. Three different network entities can be identified as follows:
• *Client*: an entity, which has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either individual consumers or organizations.
• *Cloud Storage Server* (CSS): an entity, which is managed by Cloud Service Provider (CSP), has significant storage computation resource to maintain the client's data.
• *Third Party Auditor* (TPA): an entity, which has expertise and capabilities that clients do not have, is trusted to assess and expose risk of cloud storage services on behalf of the clients upon request.

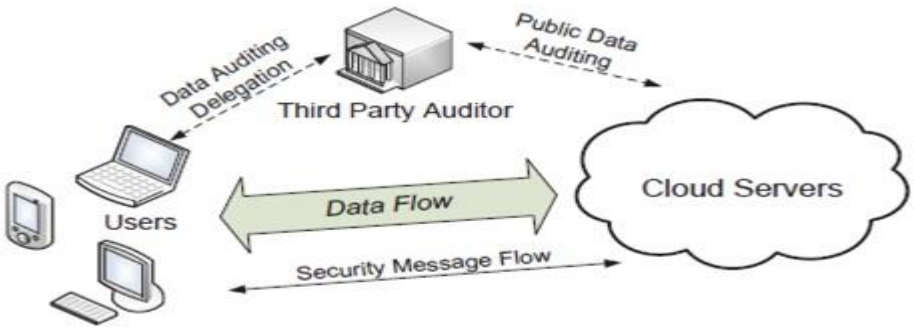

Fig. 1: Architecture of Cloud data storage service

In the cloud paradigm, by putting the large data files on the remote servers, the clients can be relieved of the burden of storage and computation. As clients no longer possess their data locally, it is of critical importance for the client's to ensure that their data are being correctly stored and maintained. That is,

clients should be equipped with certain security means so that they can periodically verify the correctness of the remote data even without the existence of local copies. In case that client does not necessarily have the time, feasibility or resources to monitor their data, they can delegate the monitoring task to a trusted TPA [5].

2.2 Existing System

In the existing system cloud data storage service involve three different entities, the cloud user (U), who has large amount of data files to be stored in the cloud; the cloud server (CS), which is managed by cloud service provider (CSP) to provide data storage service; the third party auditor (TPA), who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service security on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance (Fig 1). The TPA, who is in the business of auditing, is assumed to be reliable and independent, and thus has no incentive to collude with either the CS or the users during the auditing process. TPA should be able to efficiently audit the cloud data storage without local copy of data and without bringing in additional on-line burden to cloud users. However, any possible leakage of user's outsourced data towards TPA through the auditing protocol should be prohibited. To achieve the audit delegation and authorize CS to respond to TPA's audits, the user can sign a certificate granting audit rights to the TPA's public key, and all audits from the TPA are authenticated against such a certificate. For external attacks, data integrity threats may come from outsiders who are beyond the control domain of CSP, for example, the economically motivated attackers. They may compromise a number of cloud data storage servers in different time intervals and subsequently be able to modify or delete users' data while remaining undetected by CSP.

3. PROPOSED SYSTEM

In proposed system an effective and flexible distributed scheme with explicit dynamic data support, including block update, delete, and append. The adaptation of proposed protocol- Seb´e et al.'s protocol with distributed verification of erasure-coded data, the scheme achieves the public verifiability and data dynamics against the third party verifiers which shows the detection of data corruption during the storage correctness verification across the distributed servers. The protocol design will achieve the following security and performance guarantee:

1) Public auditability **2)** Storage correctness **3)** Privacy-preserving **4)** Batch auditing **5)** Lightweight.

The model we propose aims to protect cloud data against untrusted service providers. This model involves data owners, cloud service providers, and data users. Data owners store data in the cloud and send every share of data entries to the service providers. Data users access data from the service providers and have access to the Public information of data owners in order to verify the shares received from service providers. Fast localization of data error is to effectively locate the malfunctioning server when data corruption has been detected.

3.1 Homomorphic Token

Homomorphic authenticators are unforgeable verification metadata generated from individual data blocks, which can be securely aggregated in such a way to assure an auditor that a linear combination of data blocks is correctly computed by verifying only the aggregated authenticator. Overview to achieve privacy-preserving public auditing, we propose to uniquely integrate the homomorphic authenticator with random mask technique [10]. In our protocol, the linear combination of sampled blocks in the server's response is masked with randomness generated by a pseudo random function (PRF).

3.2 Batch Auditing

With the establishment of privacy-preserving public auditing in Cloud Computing, TPA may concurrently handle multiple auditing delegations upon different users' requests. The individual auditing of these tasks for TPA can be tedious and very inefficient [9]. Batch auditing not only allows TPA to

perform the multiple auditing tasks simultaneously, but also greatly reduces the computation cost on the TPA side.

3.3 Data Dynamics Process

To ensure the assurance of the data we can do the operations such as append, deletion, and update. These are the dynamic data operations to be done in the cloud area by user.

3.3.1 Append Operation

We may assume that there is any size of GB (Giga Bytes) space allotted by CSP user's requirement for any application purpose. Then first, this size is calculated and compared using our technique. It is clearly mentioned in our algorithm specification. In the comparison the storage cloud area is confirmed that it does not have any data in its position for strong integrity [8]. Also the operations such as add, change, and removals by client are processed by space measurement scheme for effective identification of data integrity in cloud database. So if there is any such modification by attack then client can give assurance to the data integrity by successfully following.

3.3.2 Deletion Operation

First, we want to compare value from existing cloud server. Then, this deletion operation depends on user's attempt on his data stored in cloud server using his login operation. The following operation is performed in this deletion operation. If there is number of servers for data selection while deleting the data stored the particular storage server is considered for this. The required data can be deleted using above mentioned algorithm [11].

4 PROTOCOL ANALYSIS

A cloud storage system in which there is a client and an untrusted server is considered. The client stores her data in the server without keeping a local copy. Hence, it is of critical importance that the client should be able to verify the integrity of the data stored in the remote untrusted server. If the server modifies any part of the client's data, the client should be able to detect it and should not be detected by any third party verifier. In this case, when a third party verifier verifies the integrity of the client's data, the data should be kept private against the third party verifier. The proposed protocol is correct in the sense that the server can pass the verification of data integrity as long as both the client and the server are honest. Then the protocol is secure against the untrusted server. The protocol guarantee is that, assuming the client is honest, if and only if the server has access to the complete and uncorrupted data, it can pass the verification process successfully. Finally the protocol is private against third party verifiers. To design the remote data integrity checking, Seb´e et al.'s protocol the following five functions needed are

(a) KeySetUp,
(b) TagGen,
(c) Challenge
(d) Gen-Proof
(e) Verify-Proof

Let m be the file that will be stored in the untrusted server, which is divided into n blocks of equal lengths: m = m1,m2...mn, where n = ⌈|m|/l⌉. Here l is the length of each file block.

4.1. KeySetUp(1^j) -> (pj, sj). Given the security parameter j, this function generates the public key pj and the secret key sj. pj is public to everyone, while sj is kept secret by the client.

4.2. TagGen (pj, sj, m) -> Dm. Given pj, sj and m, this function computes a verification tag Dm and makes it publicly known to everyone. This tag will be used for public verification of data integrity.

Ciphertext (c) = m^emodn

4.3. Challenge (pj, Dm) -> chal. Using this function, the verifier generates a challenge chal to request for the integrity proof of file m. The verifier sends chal to the server.

R= {(c_1, c_2, ----, c_n)modn)}

4.4. GenProof (pj, D_m, m, chal) -> R. Using this function, the server computes a response R to the challenge chal. The server sends R back to the verifier. Server uses the Homomorphic Token for collecting required information.

R'= {(m_1, m_2, -----, m_n)^{e}modn)}

4.5. VerifyProof (pj, Dm, chal, R) -> {"Verified", "Not Verified". The verifier checks the validity of the response R. If it is valid, the function outputs "Verified," otherwise the function outputs "Not verified".

5 CONCLUSION

Data Security is important and challenging issue in Cloud Computing. In proposed paper, we propose a new data integrity checking protocol for cloud storage. Already various remotely data integrity checking protocol has been developed. The proposed protocol is suitable for providing integrity protection of customer's important data. The proposed protocol supports data dynamic operations, batch auditing and also supports public verifiability. The proposed protocol is proved to be secure against an untrusted server. It also provides security against Third Party Auditor (TPA). Both theoretical analysis and experimental results demonstrate that the proposed protocol has very good efficiency in the aspects of communication, computation, and storage costs.

Following objectives have been met satisfactorily which are stated below:

- Providing secure communication between client and server using strong authentication.
- The Tag Generation time and File verification time of data files are reduced.
- Data Integrity auditing problem is optimized.

6 FUTURE SCOPE

1 In future, data locking protocols are used for reducing the modification in data files.

2 In this work, after searching a keyword user gets resultant file information in sequential order, but in ranking order searching will be provided.

REFERENCES

[1] Jeff Naruchitparames and Mehmet Hadi G¨unes, "Enhancing Data Privacy and Integrity in the Cloud"

[2] Tariq Hamid Shareef and Prof. Praveen Sen, "Privacy-Preserving Public Auditing For Secure Cloud Storage" in Proceeding of International Journal of Innovative Technologies, Vol. 2(3), pp. 121-127, Mar 2014.

[3] K.D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Proc. ACM Conf. Computer and Comm. Security (CCS '09), pp. 187-198, 2009.

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

[5] Remote Data Integrity Checking in Cloud Computing

[6] Zhuo Hao, Sheng Zhong, Member, IEEE, and Nenghai Yu, Member, "A Privacy-Preserving Remote Data Integrity Checking Protocol with Data Dynamics and Public Verifiability", IEEE-2011

[7] Cong Wang, Student Member, IEEE, Qian Wang, Student Member, IEEE, Kui Ren, Member, IEEE,Ning Cao, Student Member, IEEE, and Wenjing Lou, Senior Member, "Towards Secure and Dependable Storage Services in Cloud Computing" IEEE-2011

[8] T. Fraser, "Lomac: Low water-mark integrity protection for cots environments," in IEEE Symposium on Security and Privacy. IEEE Computer Society, 2000, pp. 230–245.

[9] Zhuo Hao, Sheng Zhong, Member, IEEE, and Nenghai Yu, Member, "A Privacy-Preserving Remote Data Integrity Checking Protocol with Data Dynamics and Public Verifiability", IEEE-2011.

[10] Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in InfoCom2010, IEEE, March 2010.

[11] Cong Wang, Member, IEEE, Sherman S.M. Chow, Qian Wang, Member, IEEE, Kui Ren, Senior Member, IEEE, and Wenjing Lou, Senior Member, IEEE "Privacy-Preserving Public Auditing for Secure Cloud Storage", in Proceeding of IEEE International Conference on System, Vol. 29(7), 2012.

