Effective Approach for Secure Cloud Storage Through Public Auditing

R. Kaviya[1] **A. Sudha**[2]
[1]P.G Scholar [2]Assistant professor
[1, 2]AL-Ameen Engineering College, Erode.

Abstract--- Cloud computing has been envisioned as the de-facto solution to the rising storage costs of IT Enterprises. With the high costs of data storage devices as well as the rapid rate at which data is being generated it proves costly for enterprises or individual users to frequently update their hardware. Apart from reduction in storage costs data outsourcing to the cloud also helps in reducing the maintenance. Cloud storage moves the user's data to large data centers, which are remotely located, on which user does not have any control. However, this unique feature of the cloud poses many new security challenges which need to be clearly understood and resolved. We provide a scheme which gives a proof of data integrity in the cloud which the customer can employ to check the correctness of his data in the cloud. This proof can be agreed upon by both the cloud and the customer and can be incorporated in the SLA.

I. INTRODUCTION

cloud storage is an important service of cloud computing, which allows data owners (owners) to move data from their local computing systems to the cloud. More and more owners start to store the data in the cloud. However, this new paradigm of data hosting service also introduces new security challenges. Owners would worry that the data could be lost in the cloud. This is because data loss could happen in any infrastructure, no matter what high degree of reliable measures cloud service providers would take, Sometimes, cloud service providers might be dishonest. They could discard the data that have not been accessed or rarely accessed to save the storage space and claim that the data are still correctly stored in the cloud. Therefore, owners need to be convinced that the data are correctly stored in the cloud. Traditionally, owners can check the data integrity based on two-party storage auditing protocols.

In cloud storage system, however, it is inappropriate to let either side of cloud service providers or owners conduct such auditing, because none of them could be guaranteed to provide unbiased auditing result. In this situation, third-party auditing is a natural choice for the storage auditing in cloud computing. A third party auditor (auditor) that has expertise and capabilities can do a more efficient work and convince both cloud service providers and owners. For the third-party auditing in cloud storage systems, there are several important requirements that have been proposed in some previous works.

The auditing protocol should have the following properties:

1) Confidentiality. The auditing protocol should keep owner's data confidential against the auditor.
2) Dynamic auditing. The auditing protocol should support the dynamic updates of the data in the cloud.
3) Batch auditing. The auditing protocol should also be able to support the batch auditing for multiple owners and multiple clouds.

Recently, several remote integrity checking protocols were proposed to allow the auditor to check the data integrity on the remote server.

In [13], the authors proposed a dynamic auditing protocol that can support the dynamic operations of the data on the cloud servers, but this method may leak the data content to the auditor because it requires the server to send the linear combinations of data blocks to the auditor. In [14], the authors extended their dynamic auditing scheme to be privacy preserving and support the batch auditing for multiple owners. However, due to the large number of data tags, their auditing protocols may incur a heavy storage overhead on the server. In [12], Zhu et al. proposed a cooperative provable data possession scheme that can support the batch auditing for multiple clouds and also extend it to support the dynamic auditing in [10]. However, their scheme cannot support the batch auditing for multiple owners. That is because parameters for generating the data tags used by each owner are different, and thus, they cannot combine the data tags from multiple owners to conduct the batch auditing. Another drawback is that their scheme requires an additional trusted organizer to send a commitment to the auditor during the multi cloud batch auditing, because their scheme applies the mask technique to ensure the data privacy. However, such additional organizer is not practical in cloud storage systems.

In this paper, we propose an efficient and secure dynamic auditing protocol, which can meet the above listed requirements. To solve the data privacy problem, our method is to generate an encrypted proof with the challenge stamp by using the Bilinearity property of the bilinear pairing, such that the auditor cannot decrypt it but can verify the correctness of the proof.

Our original contributions can be summarized as follows:

1) We design an auditing framework for cloud storage systems and propose a privacy-preserving and efficient storage auditing protocol. Our auditing protocol ensures the data privacy by using cryptography method and the Bilinearity property of the bilinear pairing, instead of using the mask technique. Our auditing protocol incurs less communication cost between the auditor and the server. It also reduces the computing loads of the auditor by moving it to the server.
2) We extend our auditing protocol to support the data dynamic operations, which is efficient and provably secure in the random oracle model.

II. SYSTEM MODELS

We consider an auditing system for cloud storage as shown in Fig. 1, which involves data owners (owner), the cloud server (server), and the third-party auditor (auditor). The owners create the data and host their data in the cloud. The cloud server stores the owners' data and provides the data access to users (data consumers). The auditor is a trusted third-party that has expertise and capabilities to provide data

storage auditing service for both the owners and servers. The auditor can be a trusted organization managed by the government, which can provide unbiased auditing result for both data owners and cloud servers.

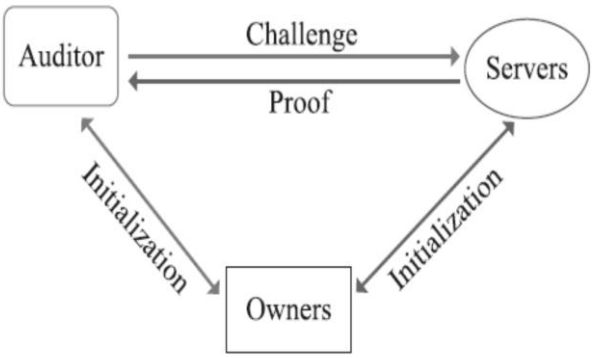

Fig. 1: System model for data storage.

A. *Existing System*

As data generation is far outpacing data storage it proves costly for small firms to frequently update their hardware whenever additional data is created. Also maintaining the storages can be a difficult task. It transmitting the file across the network to the client can consume heavy bandwidths. The problem is further complicated by the fact that the owner of the data may be a small device, like a PDA (personal digital assist) or a mobile phone, which have limited CPU power, battery power and communication bandwidth.

1) Disadvantages

- The main drawback of this scheme is the high resource costs it requires for the implementation.
- Also computing hash value for even a moderately large data files can be computationally burdensome for some clients (PDAs, mobile phones, etc).
- Data encryption is large so the disadvantage is small users with limited computational power (PDAs, mobile phones etc.).

III. MODULES

A. *Third Party Auditor*

In this module, Auditor views the all user data and verifying data .Auditor directly views all user data without key. Admin provided the permission to Auditor. After auditing data, store to the cloud

B. *Cryptography*

The art of protecting information by transforming it (encrypting it) into an unreadable format, called cipher text. Only those who possess a secret key can decipher (or decrypt) the message into plain text. Encrypted messages can sometimes be broken by cryptanalysis, also called code breaking, although modern cryptography techniques are virtually unbreakable.

C. *Cloud Computing*

Cloud computing is the provision of dynamically scalable and often virtualized resources as a services over the internet Users need not have knowledge of, expertise in, or control over the technology infrastructure in the "cloud" that supports them. Cloud computing represents a major change in how we store information and run applications. Instead of hosting apps and data on an individual desktop computer, everything is hosted in the "cloud" an assemblage of computers and servers accessed via the Internet.

Cloud computing exhibits the following key characteristics:

1) Agility

It improves with users' ability to re-provision technological infrastructure resources.

2) Multi tenancy

It enables sharing of resources and costs across a large pool of users thus allowing for

3) Utilization and efficiency

Its improvements for systems that are often only 10–20% utilized.

4) Reliability

It is improved if multiple redundant sites are used, which makes well-designed cloud computing suitable for business continuity and disaster recovery.

5) Performance

It is monitored and consistent and loosely coupled architectures are constructed using web services as the system interface.

6) Security

It could improve due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Security is often as good as or better than other traditional systems, in part because providers are able to devote resources to solving security issues that many customers cannot afford. However, the complexity of security is greatly increased when data is distributed over a wider area or greater number of devices and in multi-tenant systems that are being shared by unrelated users. In addition, user access to security audit logs may be difficult or impossible. Private cloud installations are in part motivated by users' desire to retain control over the infrastructure and avoid losing control of information security.

7) Maintenance

Maintenance of cloud computing applications is easier, because they do not need to be installed on each user's computer and can be accessed from different places.

D. *Simply Archives*

This problem tries to obtain and verify a proof that the data that is stored by a user at remote data storage in the cloud (called cloud storage archives or simply archives) is not modified by the archive and thereby the integrity of the data is assured. Cloud archive is not cheating the owner, if cheating, in this context, means that the storage archive might delete some of the data or may modify some of the data. While developing proofs for data possession at untrusted cloud storage servers we are often limited by the resources at the cloud server as well as at the client.

E. *Sentinels*

In this scheme, unlike in the key-hash approach scheme, only a single key can be used irrespective of the size of the file or the number of files whose retrievability it wants to verify. Also the archive needs to access only a small portion of the file F unlike in the key-has scheme which required the archive to process the entire file F for each protocol verification. If the prover has modified or deleted a

substantial portion of F, then with high probability it will also have suppressed a number of sentinels.

F. Verification Phase

The verifier before storing the file at the archive, preprocesses the file and appends some Meta data to the file and stores at the archive. At the time of verification the verifier uses this Meta data to verify the integrity of the data. It is important to note that our proof of data integrity protocol just checks the integrity of data i.e. if the data has been illegally modified or deleted. It does not prevent the archive from modifying the data.

G. Advantages

We motivate the public auditing system of data storage security in Cloud Computing, and propose a protocol supporting for fully dynamic data operations, especially to support block insertion, which is missing in most existing schemes;

We extend our scheme to support scalable and efficient public auditing in Cloud Computing. In particular, our scheme achieves auditing tasks from different users can be performed simultaneously by the TPA.

H. Architecture of Cloud Computing

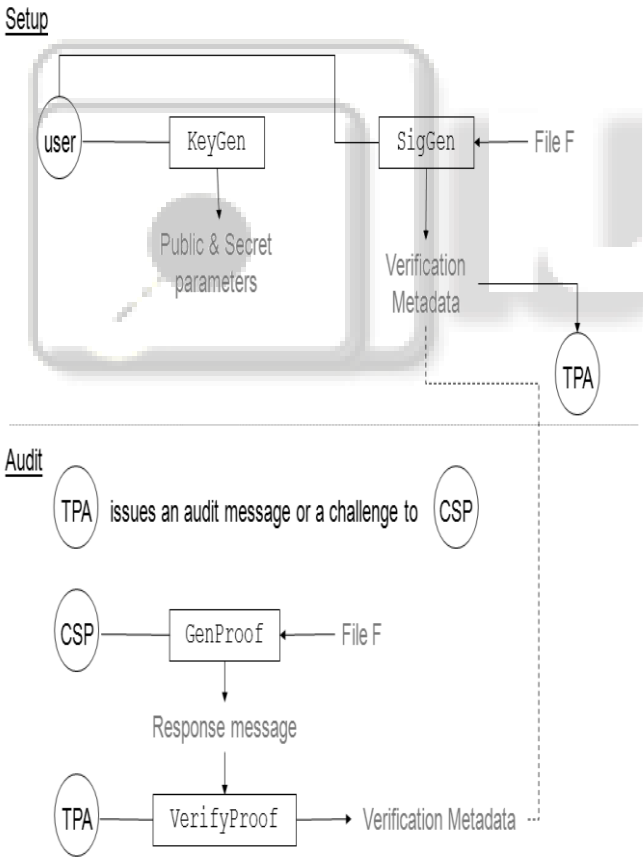

Fig. 3: flow diagram

To enable privacy-preserving public auditing for cloud data storage under the aforementioned model, our protocol design should achieve the following security and performance guarantee

1) Public auditability: to allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional on-line burden to the cloud users.
2) Storage correctness: to ensure that there exists no cheating cloud server that can pass the audit from TPA without indeed storing users' data intact.
3) Privacy-preserving: to ensure that there exists no way for TPA to derive users' data content from the information collected during the auditing process.
4) Batch auditing: to enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously.
5) Lightweight: to allow TPA to perform auditing with minimum communication and computation overhead. Fig 3 shows the flow diagram of this paper.

IV. CONCLUSION

In this paper we have worked to facilitate the client in getting a proof of integrity of the data which he wishes to store in the cloud storage servers with bare minimum costs and efforts. Our scheme was developed to reduce the computational and storage overhead of the client as well as to minimize the computational overhead of the cloud storage server. We also minimized the size of the proof of data integrity so as to reduce the network bandwidth consumption. Many of the schemes proposed earlier require the archive to perform tasks that need a lot of computational power to generate the proof of data integrity. But in our scheme the archive just need to fetch and send few bits of data to the client.

REFERENCES

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," technical report, Nat'l Inst. of Standards and Technology, 2009.
[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," Comm. ACM, vol. 53, no. 4, pp. 50-58, 2010.
[3] T. Velte, A. Velte, and R. Elsenpeter, Cloud Computing: A Practical Approach, first ed., ch. 7. McGraw-Hill, 2010.
[4] J. Li, M.N. Krohn, D. Mazie`res, and D. Shasha, "Secure Untrusted Data Repository (SUNDR)," Proc. Sixth Conf. Symp. Operating Systems Design Implementation, pp. 121-136, 2004.
[5] G.R. Goodson, J.J. Wylie, G.R. Ganger, and M.K. Reiter, "Efficient Byzantine-Tolerant Erasure-Coded Storage," Proc. Int'l Conf. Dependable Systems and Networks, pp. 135-144, 2004.
[6] V. Kher and Y. Kim, "Securing Distributed Storage: Challenges, Techniques, and Systems," Proc. ACM Workshop Storage Security and Survivability (StorageSS), V. Atluri, P. Samarati, W. Yurcik, L. Brumbaugh, and Y. Zhou, eds., pp. 9-25, 2005.
[7] L.N. Bairavasundaram, G.R. Goodson, S. Pasupathy, and J. Schindler, "An Analysis of Latent Sector Errors in Disk Drives," Proc. ACM SIGMETRICS Int'l Conf. Measurement and Modeling of Computer Systems, L.

Golubchik, M.H. Ammar, and M. Harchol- Balter, eds., pp. 289-300, 2007.

[8] B. Schroeder and G.A. Gibson, "Disk Failures in the Real World: What Does an MTTF of 1,000,000 Hours Mean to You?" Proc. USENIX Conf. File and Storage Technologies, pp. 1-16, 2007.

[9] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard, "A Cooperative Internet Backup Scheme," Proc. USENIX Ann. Technical Conf., pp. 29-41, 2003.

[10] Y. Deswarte, J. Quisquater, and A. Saidane, "Remote Integrity Checking," roc. Sixth Working Conf. Integrity and Internal Control in Information Systems (IICIS), Nov. 2004.

[11] M. Naor and G.N. Rothblum, "The Complexity of Online Memory Checking," J. ACM, vol. 56, no. 1, article 2, 2009.

[12] A. Juels and B.S. Kaliski Jr., "Pors: Proofs of Retrievability for Large Files," Proc. ACM Conf. Computer and Comm. Security, P. Ning, S.D.C. di Vimercati, and P.F. Syverson, eds., pp. 584-597, 2007.

[13] T.J.E. Schwarz and E.L. Miller, "Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage," Proc. 26th IEEE Int'l Conf. Distributed Computing Systems, p. 12, 2006.

[14] D.L.G. Filho and P.S.L.M. Barreto, "Demonstrating Data Possession and Uncheatable Data Transfer," IACR Cryptology ePrint Archive, vol. 2006, p. 150, 2006.

