Ensuring Data Integrity by Using Secure Audit Service Model in Cloud

[1]Syed Ameenulla, [2]K. Navin

1PG Scholar, [2]Assistant Professor
[1]Department of IT, SRM University, Chennai, India
[1]Syedameen1201@gmail.com, [2]Navin.k@ktr.srmuniv.ac.in

***Abstract*—Cloud based outsourced storage reduces the client's burden for storage management and maintenance by providing a comparatively low cost, scalable, location independent platform. However there are various existing audit services for data integrity but it is not possible to use them. Since the data stored in the cloud is maintained remotely. It requires efficient and secure audit service for maintaining data integrity in cloud. However the clients no longer need to face various security risks like corrupted or missing data. To avoid the security risks of data, audit services are critical to ensure integrity of outsourced data and to achieve credibility and digital forensics on cloud computing. Provable Data possession (PDP) is a cryptographic technique is used to verify the data integrity without retrieving it at an untrusted server. In this paper, we address the construction of an interactive PDP Protocol to prevent fraudulence of prover and the leakage of data.**

***Index Terms*—PDP, Cryptographic Technique, Audit Service**

I. INTRODUCTION

In recent years, the emerging cloud-computing paradigm is rapidly gaining momentum as an alternative to traditional information technology. Cloud computing provides a scalability environment for growing amounts of data and processes that work on various applications and services by means of on-demand services. One fundamental aspect of this paradigm shifting is that data are being centralized and outsourced into clouds. This kind of storage services in clouds have become a new profit growth point by providing a comparably scalable, low-cost location-independent platform for managing clients' data.

The cloud storage service (CSS) relieves the burden of storage management and its maintenance. However, if such an important service is vulnerable to attacks or failures, then it would bring irretrievable losses to users since their data or archives are stored into an uncertain storage pool outside the enterprises. These security problems come from the following reasons: the cloud infrastructures are much more powerful and reliable than personal computing devices.

However, they are still susceptible to security threats both from outside and inside the cloud (Armbrust et al., 2010); for the benefits of their tenture, there exist various motivations for cloud service providers (CSP) to behave unfaithfully toward the cloud users furthermore, the dissension occasionally suffers from the lack of trust on CSP. Therefore, their behaviors may not be known by the cloud users, even if this dissension may result from the users' own improper operations. Therefore, it is necessary for CSP to offer an efficient audit service to check the integrity and availability of the stored data .

Traditional cryptographic technologies for data integrity and availability, based on signature schemes and hash functions (Hsiaoet al., 2009; Yumerefendi and Chase, 2007), cannot work on the outsourced data without a local copy of data. In addition, it is not a practical solution for data validation by downloading them due to the expensive transaction, especially for large-size files. Moreover, the solutions to audit the correctness of the data in a cloud environment can be formidable and expensive for the cloud users. Therefore, it is crucial to realize public audit ability for CSS, in case the data owners may resort to a third party auditor (TPA), who has expertise and capabilities that a common user does not have, for periodically auditing the outsourced data. This audit service is significantly important for digital forensics and data assurance in clouds.

Although PDP/POR schemes evolved around untrusted storage offer a publicly accessible remote interface to check and manage tremendous amount of data, most of existing schemes cannot give a strict security proof against the untrusted CSP's deception and forgery, as well as information leakage of verified data in verification process. These drawbacks greatly affect the impact of cloud audit services. Thus, new frameworks or models are desirable to enable the security of public verification protocol in cloud audit services.

Another major concern addressed by this paper is how to improve the performance of audit services. The audit performance concerns not only the costs of computation, communication, storage for audit activities but also the scheduling of audit activities. No doubt improper scheduling, more or less frequent, causes poor audit performance, but a critical scheduling can help provide a better quality of and a more cost-effective service. Hence, it is critical to investigate an efficient schedule for cloud audit services.

In response to practical requirements for outsourced storages, our concerns to improve the performance of audit services are mainly from three aspects:

- How to design an efficient architecture of audit system to reduce the storage and network overheads and enhance the security of audit activities;
- How to provide an efficient audit scheduling to help provide a more cost-effective audit service;

- How to optimize parameters of audit systems to minimize the computation overheads of audit services.

Solving these problems will help to improve the quality of audit services, which can not only timely detect abnormality, but also take up less resource, or rationally allocate resources.

II. RELATED WORK

There has been a considerable amount of work done on untrusted outsourced storage. The most direct way to enforce the integrity control is to employ cryptographic hash function. Yumerefendi and Chase proposed a solution for authenticated network storage (Yumerefendi and Chase, 2007; Hsiao et al., 2009), using a hash tree (called as Merkle tree) as the underlying data structure. However their processing of updates is computationally expensive. Fu et al. (2002) described and implemented a method for efficiently and securely accessing a read-only file system that has been distributed to many providers. This architecture is a solution for efficiently authenticating operations on an outsourced file system.

Some recent work (Li et al., 2006; Ma et al., 2005; Xie et al., 2007; Yavuz and Ning, 2009) studied the problem of auditing the integrity for outsourced data or database. By explicitly assuming an order of the records in database, Pang et al. (Ma et al., 2005) used an aggregated signature to sign each record with the information from two neighboring records in the ordered sequence, which ensures the result of a simple selection query is continuous by checking the aggregated signature. Other work (Li et al., 2006; Xie et al., 2007) used a Merkle tree to audit the completeness of query results, but in some extreme cases, the overhead could be as high as processing these queries locally, which can significantly undermine the benefits of database outsourcing. Moreover, to ensure freshness, an extra system is needed to deliver the up-to-date root signature to all clients in a reliable and timely manner.

III. AUDIT SYSTEM ARCHITECTURE

In this section, we first introduce an audit system architecture for outsourced data in clouds in Fig. 1, which can work in an audit service outsourcing mode. In this architecture, we consider a data storage service containing four entities:.

A. Data Owner

Who is going to upload the data in the cloud service provider. The data owner needs to register with the CSP to allot the space.

B. Cloud service provider

They provide data storage service and has enough storage spaces and computation resources.

C. Third Party Auditor

Capability to manage or monitor outsourced data under the delegation of data owner.

D. Granted Applications

These applications can be either inside clouds or outside clouds according to the specific requirements. They have the right to access and manipulate stored data.

In this architecture, the clients and the data owner need to dynamically interact with the cloud service provider to access or update their data for various application purposes. In this architecture TPA is a trusted third party, is used to ensure the security of outsourced data. The users can upload or download their information in the cloud. The data owner verifies the information stored in the cloud. The data owner provides permissions for granted application. Then the users can access the outsourced data. Hence TPA is considered as a reliable and independent.

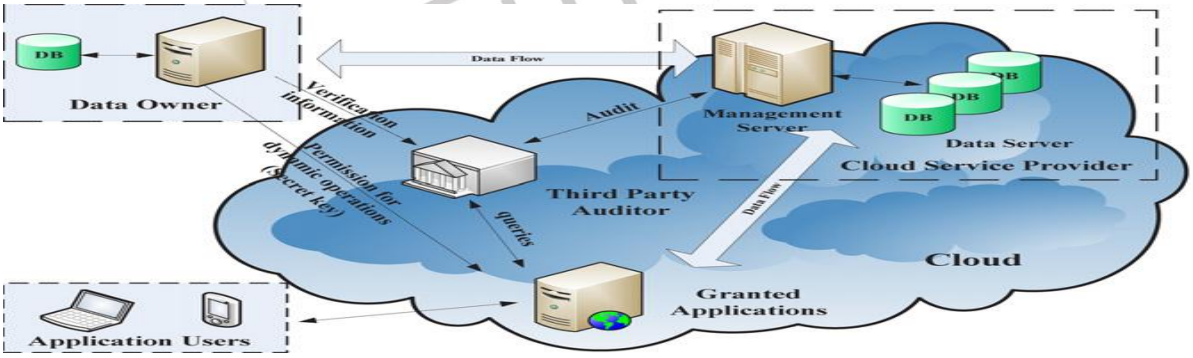

Fig.1. Audit System Architecture

- TPA should able to monitor dynamically on integrity and availability of these delegated data at appropriate intervals.
- TPA should be able to take the evidences for the disputes about the inconsistency of data in terms of authentic records for all data operations.
- TPA Provides high performance by providing minimum overheads in storage.

In this audit architecture, our core idea is to maintain the security of TPA to guarantee the credibility of cloud storages. This is because it is more easy and reliable to ensure the security of one TTP than to maintain the credibility of the whole cloud. Hence, the TPA could be considered as the root of trust in clouds.

IV. INTERACTIVE AUDITING SCHEME

In this section, we propose a cryptographic interactive audit to support our audit system in cloud. This technique is constructed on the standard model of interactive proof system, which can ensure the confidentiality of secret data and the undeceivability of invalid tags.

Notations and preliminaries

Let H = {Hk} be a keyed hash family of functions Hk :{0, 1}*→ {0, 1}n indexed by k ∈ K. We say that algorithm A has advantage _ in breaking the collision-resistance of H if

Pr[A(k) = (m0, m1) : m0 / = m1, Hk(m0) = Hk(m1)] ≥ _,

where the probability is over random choice of k ∈ K and random bits of A. This hash function can be obtained from hash function of BLS signatures.

Definition 1 (Collision-resistant hash). A hash family H is (t, _)- collision-resistant if no t-time adversary has advantage at least _ in breaking the collision-resistance of H.

We set up our systems using bilinear pairings proposed by Boneh and Franklin (2001). Let G be two multiplicative groups using elliptic curve conventions with large prime order p. The function e be a computable bilinear map e : G × G → GT with following properties: for any G, H ∈ G and all a, b ∈ Zp, we have (1) Bilinearity: e([a]G, [b]H) = e(G, H)ab. (2) Non-degeneracy: e(G, H) / = 1 unless G or H = 1. (3) Computability: e(G, H) is efficiently computable.

Definition 2 (Bilinear map group system). A bilinear map group system is a tuple S = _p, G, GT , e_ composed of the objects as described above.

Definition 3. A cryptographic interactive audit scheme S is a collection of two algorithms and an interactive proof system, S = (K,T,P):

1. KeyGen(1s)
takes a security parameter s as input, and returns a public-secret keypair (pk, sk);

2. TagGen(sk, F)
takes as inputs the secret key sk and a file F, and returns the triples (_, , _), where _ denotes the secret used to generate verification tags, is the set of public verification parameters u and index information _, i.e., = (u, _); _ denotes the set of verification tags;

***3.* Proof (CSP, TPA):**
is a public two-party proof protocol of retrievability
between CSP (prover) and TPA (verifier), that is _CSP(F, _), TPA_(pk,), where CSP takes as input a file F and a set of tags _, and a public key pk and a set of public parameters are the common input between CSP and TPA. At the end of the protocol run, TPA returns {0|1}, where 1 means the file is correct stored on the server. where, P(x) denotes the subject P holds the secret x and _P, V_(x) denotes both parties P and V share a common data x in a protocol.

V. PERFORMANCE EVALUATION

The audit service achieves the detection of CSP servers misbehavior in a random sampling mode in order to reduce the workload on the server. The detection probability P of disrupted blocks is an important parameter to guarantee that these blocks can be detected in time. Assume the TPA modifies e blocks out of the n-block file. The probability of disrupted blocks is _b = e/n. Let t be the number of queried blocks for a challenge in the protocol proof. We have detection probability

$$P = 1 - \left(\frac{n-e}{n}\right)^t = 1 - (1-\rho_b)^t.$$

Hence, the number of queried blocks is t = log (1−P)/log (1−Pb)2.

In the below Fig. We show the same result for the number of queried blocks under different detection probabilities (from 0.5 to 0.99), the different number of file blocks (from 200 to 10,000), and the constant number of disrupted blocks (100). It is easy to find that the number of queried blocks t is directly proportional to the total number of file blocks n for the constant P and e, that is, t ≈ c(P n)/e for a sufficiently large n, where c is a constant.

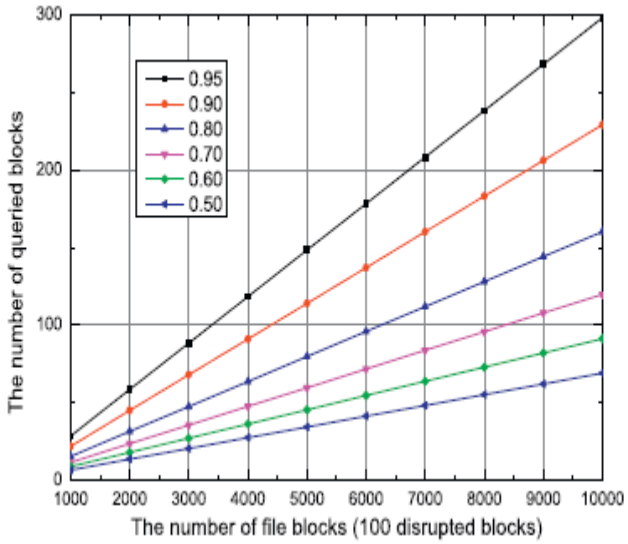

Fig.2. Number of queried blocks under different direction probabilities and the different number of file blocks

Furthermore, we observe the ratio of queried blocks in the total file blocks w = t/n under different detection probabilities. Based on above analysis, it is easy to find that this ratio holds the equation.

$$w = \frac{t}{n} = \frac{\log(1-P)}{n \cdot \log(1-\rho_b)}.$$

In most cases, we adopt the probability of disrupted blocks to describe the possibility of data loss, damage, forgery or unauthorized changes. When this probability _b is a constant probability, the TPA can detect sever misbehaviors with a certain probability P by asking proof for a constant amount of blocks t = log (1 − P)/log (1 − _b), independently of the total number of file blocks (Ateniese et al., 2007). In the above Fig. we show the ratio changes for different detection probabilities under 1% disrupted blocks, e.g.,the TPA asks for 458, 298 and 229 blocks in order to achieve P of at least 99%, 95% and 90%, respectively. This kind of constant ratio is useful for the uniformly distributed _b, especially for the storage device's physical failures.

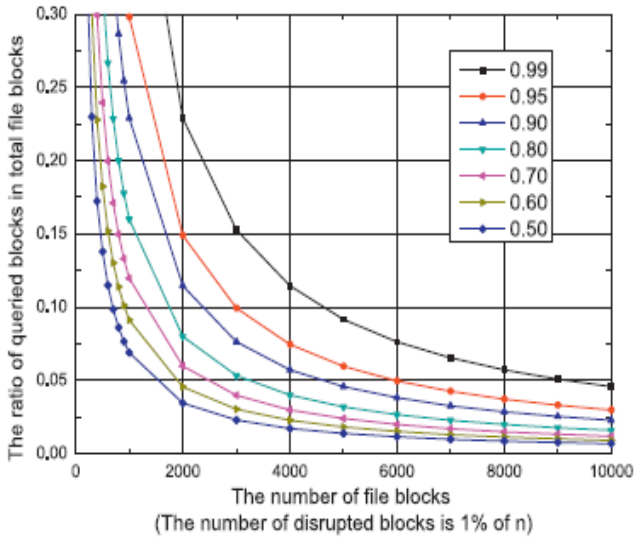

Fig. 3. Ration of queried blocks

VI. CONCLUSION

In this paper, we addressed the construction of an efficient secure auditing system for data integrity and data availability in cloud. We can implement effective system for user by using Third party Auditor. By using this system we can reduce the workload on cloud by maintaining security of TPA. The users can use the cloud storage service without any risk. It is easy to deploy in cloud and provides good performance than the traditional hash based solution. Since We propose an efficient and reliable secure audit service outsourcing for data integrity in cloud system.

ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression, "One of us (R. B. G.) thanks . . ." Instead, try "R. B. G. thanks". Put applicable sponsor acknowledgments here; DO NOT place them on the first page of your paper or as a footnote.

REFERENCES

[1] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M., 2010. A view of cloud computing. Commun. ACM 53 (4), 50–58.

[2] Ateniese, G., Burns, R.C., Curtmola, R., Herring, J., Kissner, L., Peterson, Z.N.J., Song, D.X., 2007. Provable data possession at untrusted stores. In: Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, pp. 598–609.

[3] Ateniese, G., Pietro, R.D., Mancini, L.V., Tsudik, G., 2008. Scalable and efficient provable data possession. In: Proceedings of the 4th International Conference on Security and Privacy in Communication Networks, SecureComm, pp. 1–10.

[4] Barreto, P.S.L.M., Galbraith, S.D., O'Eigeartaigh, C., Scott, M., 2007. Efficient pairing computation on supersingular abelian varieties. Des. Codes Cryptogr. 42 (3), 239–271.

[5] Beuchat, J.-L., Brisebarre, N., Detrey, J., Okamoto, E., 2007. Arithmetic operators for pairing-based cryptography. In: Cryptographic Hardware and Embedded Systems – CHES 2007, 9th International Workshop, pp. 239–255.

[6] Boneh, D., Boyen, X., Shacham, H.,2004. Short group signatures. In: In Proceedings of CRYPTO 04, LNCS Series. Springer-Verlag, pp. 41–55.

[7] Boneh, D., Franklin, M., 2001. Identity-based encryption from the weil pairing. In:Advances in Cryptology (CRYPTO'2001). Vol. 2139 of LNCS, pp. 213–229.

[8] Bowers, K.D., Juels, A., Oprea, A., 2009. Hail: a high-availability and integrity layer for cloud storage. In: ACM Conference on Computer and Communications Security, pp. 187–198.

[9] X. Li, H. Zhang and Y. Zhang, Deploying Mobile Computation in Cloud Service , Lecture Notes in Computer Science, Vol. 5931, Cloud Computing, pp. 301-311, 2009.

[10] M. Satyanarayanan, P. Bahl, R. C´aceres, and N. Davies, The Case for VM-Based Cloudlets in Mobile Computing, IEEE Pervasive Computing, vol. 8, no. 4, pp. 14–23, Oct. 2009.

[11] Wang, C., Wang, Q., Ren, K., Lou, W., 2010. Privacy-preserving public auditing for data storage security in cloud computing. In: INFOCOM, 2010 Proceedings IEEE, pp. 1–9, 14-19.

[12] Wang, Q., Wang, C., Li, J., Ren, K., Lou, W., 2009. Enabling public verifiability and data dynamics for storage security in cloud computing. In: Proceedings of the 14th European Symposium on Research in Computer Security, ESORICS 2009, pp. 355–370.

[13] Xie, M., Wang, H., Yin, J., Meng, X., 2007. Integrity auditing of outsourced data. In: Koch, C., Gehrke, J., Garofalakis, M.N., Srivastava, D., Aberer, K., Deshpande, A., Florescu, D., Chan, C.Y., Ganti, V., Kanne, C.-C., Klas, W., Neuhold, E.J. (Eds.), VLDB. ACM, pp. 782–793.

[14] Yavuz, A.A., Ning, P., 2009. Baf: An efficient publicly verifiable secure audit logging scheme for distributed systems. In: ACSAC, pp. 219–228.

[15] Yumerefendi, A.R., Chase, J.S., 2007. Strong accountability for network storage. ACM Trans. Storage (TOS) 3 (3).

[16] Hsiao, H.-C., Lin, Y.-H., Studer, A., Studer, C., Wang, K.-H., Kikuchi, H., Perrig, A., Sun, H.-M., Yang, B.-Y., 2009. A study of user-friendly hash comparison schemes. In: ACSAC, pp. 105–114.

[17] Li, F., Hadjieleftheriou, M., Kollios, G., Reyzin, L., 2006. Dynamic authenticated index structures for outsourced databases. In: Chaudhuri, S., Hristidis, V., Polyzotis, N. (Eds.), SIGMOD Conference. ACM, pp. 121–132.