International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 03 Issue: 01 | Jan-2016 www.irjet.net p-ISSN: 2395-0072

Development of Effective Audit Service to Maintain Integrity of Migrated Data in Cloud

Shekhar S. Kausalye[1], Prof. P.N. Kalavadekar[2]

1 PG Student, Dept. of Computer Engineering, SRES COE, Kopargaon, Pune University, MH, INDIA
[2] Associate Professor, Dept. of Computer Engineering, SRES COE, Kopargaon, Pune University, MH, INDIA

Abstract - ***Cloud based service for storage of outsource data reduces the burden of user for storage management & maintenance, this is achieved because cloud service provides low-cost, expandable & geographically location independent platform. But client has risk of missing or corrupted data as the local copy is not available at client side. To avoid this security risk audit service must ensure the integrity of data. PDP technique which is used for verifying the integrity of data can be used to understand the audit service. A PDP protocol can be constructed using the interactive zero knowledge system to prevent the fraud of prover as well as leakage of verified data. This can be proved that the protocol can persists these properties with the help of Diffie-Hellman assumption. The mechanism can be proposed to reduce cost of audit and to detect misbehaviour with the help of related queries & periodic verification.***

Key Words: *security, interactive proof system, cloud storage, Audit Service, Provable Data Possession, etc…*

1. INTRODUCTION

Cloud computing is growing service for current information technology. IT provides scalable environment for ever increasing amount of data & processes which work on various application & services by using on-demand self-services. The important advantage of cloud computing paradigm is that data cam be outsourced and centralized at different location called as clouds. This kind of outsourced data storage service is becoming profitable business because of flexible, low-cost, location-independent platform to manage user data.

The burden of storage management & maintenance is reduced by Cloud Storage Service (CSS). But if this important service is susceptible to failure of attacks then this will be causing the irretrievable loss of user data as this data is not available locally with user. This risk can be caused due to: cloud infrastructure is vulnerable to security threats from within and outside of cloud [2]; the Cloud Service Providers (CSP) can behave untruthfully for cloud users [3]; also, the different opinion may be there because of faithfulness of CSP. Accordingly, their behaviour is unknown for different clients even though this may result from user's incorrect actions [4]. Because of all these reasons its cloud service provider responsibility to issue an efficient audit service so that the integrity and data availability can be verified on cloud data [5].

There are various traditional cryptographic technologies are available for data integrity and availability which are based on hash function and signature schemes [6][7]. But these technology does not work on outsourced data as they must have the local copy of data to perform the audit. As well as this process is not feasible to perform the validation by downloading data because of expensive transaction if file size is large. Also to perform the audit in cloud environment is difficult and costly for the users [8]. Therefore, it is important to understand the public auditability for CSS, because of which users may use or take help of Third Party Auditor (TPA). This TPA is expert and has capabilities that user don't have to perform the audit on outsourced data.

For this auditability the concept of Proof of Retrievability [9] commonly called as POR and Provable Data Possession [10] (POP) is given by researchers. Probabilistic proof techniques approach is used by them to show that user's data remain intact without downloading it, and this is called as "Verification without Downloading." Anyone can use the verification protocol to prove the availability of data using PDP or POR scheme. Hence this fulfils the requirement of public auditability. These schemes are developed around and untrusted storage which offer to check the very large amount of data.

Various existing scheme does not give the security proof against Cloud Service Provider's fraud and counterfeits as well as information leakage. These problems reduce the impact of cloud audit service. Because of this new model of framework must be proposed that enables the security of public verification protocol in audit service.

The performance of audit service is depending on the three aspects which are:

- How to design an efficient architecture of audit system that reduce the storage and network overheads and also 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.

Quality of service can be improved by solving above problems and it also help to detect irregularities and requires less resources.

2. EASE OF USE

Various researchers worked on the untrusted outsourced data. The simplest and easy way to implement the integrity control is to use cryptographic hash function. Yumerefendi [7] and Hsiao[6] proposed a solution for authenticated network storage, Merkle tree also called as hash tree as the fundamental data structure, but their update process is costly. Fu [11] implemented & described the method for securely accessing a read only file system which is distributed among the various providers. This is the solution for efficiently authenticating operation on outsourced data or file.

The work done by Li[12], Ma[13], Xie[14], Yavuz[5] considered the problem of how to audit the integrity of outsourced data or database. From these Li & Xie used the concept of Markle tree to audit the completeness of query result, but in some cases the overhead is high that weaken the benefits of database outsourcing. Some researchers proposed two schemes called PDP & POR to check the integrity of stored data without downloading it. Ateniese proposed the Provable Data Possession model to ensure the possession of file on untrusted storage and also provides the RSA based scheme that achieves O(1) communication cost. Anyone other than owner can challenge the server for data possession using the publicly verifiable version.

The dynamic PDP solution called scalable PDP [15] is developed by the Ateniese. This is achieved using the Lightweight PDP scheme which used cryptographic Hash function & symmetric key encryption, but server can victimize the user as lack of randomness in the challenge in previous response. The number of updates and challenges is fixed and limited. By studying this work Erway has proposed two dynamic PDP [16] scheme using Hash function tree to cognize O(log n) communication and computation cost of data (file) which contain n blocks. The drawback of SPDP is simple scheme called DPDP-I and in the "block-less scheme' which is generally called as DPDP-II the data block can be leaked by response given to challenge.

But Juels presented the POR [9] scheme which greatly cognize of pre-processing steps of the client which are followed before sending the file or data to Cloud Service Provider. The main drawback of this scheme is that these operation restricts the updating of data. Shacham & Waters [17] presented and efficient version of this protocol which is called Compact POR, this protocol uses harmonic property to aggregate a proof into O(1) authenticator value and O(t) computation cost where t is challenge blocks, but the drawback of this is that its solution is static and there is leakage of data blocks in the verification process.

3. LITERATURE SURVEY

Audit system architecture for outsourced data in cloud is shown in Fig. 1. This architecture work in audit service outsourcing mode. This architecture contains four entities:

1. Data Owner (DO): The user who want to store large amount of data to the cloud;
2. Cloud Service Provider (CSP): CSP provides data storage service and has storage space and computation resources;
3. Third Party Auditor (TAP): TPA has capabilities to manage of monitor outsources data under the deputation of data owner;

Granted Application (GA): GA has rights to access & manipulate stored data. These application can be either inside the cloud or outside the cloud. This depends on the specific requirement.

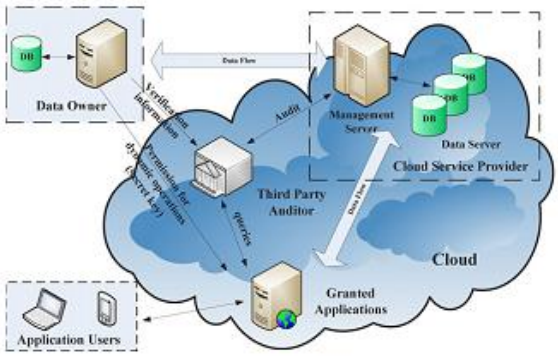

Fig-1. Audit System Architecture for Cloud Computing

Audit service based on TPA works in the following way. This also help to describe our audit service as follows:

1. Client uses the secret key sk for preprocessing of the file, file contains collection of n blocks, then it generates public verification information which is stored in TPA, after this the file is transmitted with some verification tags to Cloud Service Provider, user can then delete the local copy of this data;
2. Afterword, using protocol called as proof of retrievability, TPA check the integrity and availability of the outsourced data. It is important to generate an alarm for suspicious activity or events. [1].

This is also known as audit service outsourcing because verification of data integrity can be implemented by TPA without taking help of data owner. Here data owner and granted clients dynamically interact with Cloud Service Provider to access or update their own data for various

application purposes. However, we cannot trust CSP for security of stored data, neither can we assume that data owner is able to collect the evidences against CSP for CSPs fault when errors occur. Because of these reasons, TPA, which act as a Trust Third Party (TTP), is used to make sure the security of stored data. We can say that the TPA is reliable as well as independent, and hence it has no motivation to conspire with the CSP or clients at the time of auditing process:

1. TPA must regularly check the integrity and availability of data at appropriate intervals;
2. TPA must be able to take evidences for various disagreements about inconsistency of the data in terms of authentic records for all the data operations.

4. IMPLEMENTATION

In this proposed audit architecture, main idea is, to maintain security of Third Party Auditor so that we can guarantee the credibility of cloud storage. Because it is easy to ensure the security of Trust Third Party Auditor than maintaining the security of whole cloud. Hence, Third Party Auditor is the root or basis of trust in clouds. The protocol design must achieve following security and performance guarantees:

- Audit without downloading: Allow Third Party Auditor or clients with the help of Third Party Auditor to verify correctness of data whenever necessary without downloading the copy of data;
- Verification correctness: to ensure there is no cheating of CSP that can clear or grant the audit from Third Party Auditor without storing user's data intact;
- Privacy-preserving: to ensure that there is no alternate way for Third Party Auditor to derive user's data from information collected during the audit operation; and
- High-performance: to allow Third Party Auditor to perform auditing but with minimum overheads for storage, communication and computation [1]

In this proposed audit architecture, main idea is, to maintain security.

4.1 Design

A cryptographic interactive audit scheme is used to support the audit service. This scheme is also called as interactive PDP (IPDP). A standard model of interactive proof system is used to construct this scheme, because this ensures the confidentiality of secret data also called as zero-knowledge property & the undeceivability of invalid tags called as soundness property.

Definition Collision Resistance Hash: A hash family H is (t, ε) collision resistance if no t-time advantage at least ε in breaking the collision resistance of H [1]. The system is setup using bilinear pairing proposed by Boneh & Franklin [18]. Let $\mathbb{G}$ be two multiplicative groups using elliptic curve convention with large prime order p. A function e can be bilinear map which can be computed as: e: $\mathbb{G} \rightarrow \mathbb{G}_T$.

This has following properties: for any G, H ε $\mathbb{G}$ & all a, b ε $\mathbb{Z}_p$, we have the following –

1. Bilinearity: e([a]G; [b]H) = e(G; H) ab1:
2. Non-degeneracy: e(G; H) 6 = 1 unless G or H = 1.
3. Computability: e(G; H) is efficiently computable [1]

Definition Bilinear map group system: It is a tuple $\mathbb{S} = \langle P, \mathbb{G}, \mathbb{G}_T, e \rangle$ composed as the object as described above [1].

4.2 Algorithm

Interactive audit protocol is based on the interactive proof system as explained below: A cryptographic interactive audit scheme S is consisting of two algorithms & one proof system generally called as interactive proof system, i.e. S = (K, T, P).

Algorithm 1: KeyGen (1^s)

Security parameter s is given as an input to this algorithm, and the algorithm return a public keypair (pk,sk) as:

Step 1: Let $\mathbb{S} = (p, \mathbb{G}, \mathbb{G}_T, e)$ be a bilinear map group system with randomly selected generators g, h, ∈R,$\mathbb{G}$, where $\mathbb{G}$, $\mathbb{G}_T$ are two group of large prime order p, |p|= O(k).

Step 2: Now we generate the collision-resistance hash function $H_k(\bullet)$ and by choosing a random α, β, $\in_R$, $\mathbb{Z}_p$ and compute $H_1 = h^{\alpha}$ and $H_2 = h^{\beta} \in \mathbb{G}$.

Step 3: Thus the secret key is *sk* = (*α, β*) and the public key is *pk* = (*g,h,H1,H2*).

Algorithm 2: TagGen (sk,F)

This algorithm takes an secret key and a file F as input. This returns a triplet (ζ, ψ, σ),

where ζ denotes the secret used to generate verification tag, ψ is the set of public verification parameters u and index information χ i.e., $\psi = (u, \chi)$; σ denotes the set of verification tags;

1. Owners file i.e. *F* is splited in to *n* x *s* sectors $F=\{m_{i,j}\} \in \mathbb{Z}_p^{n \times s}$,
2. Choose *s* and random $\tau_1, \cdots, \tau_s \in \mathbb{Z}_p$ as the secret of the file.
3. Compute $u_i = g^{\tau_i} \in \mathbb{G}$ for $i \in [1, s]$ and $\xi^{(1)} = H_{\xi}(\text{“}Fn\text{”})$, where $\xi = \sum_{i=1}^{i=1}$ and F_n is the file name.
4. Build index table $\chi = \{\chi_i\}_{i=1}^{n}$.
5. Calculate its tag as: $\sigma_i \leftarrow (\xi_i^{(2)})^{\alpha} \cdot g^{\sum_{j=1}^{s} \tau_j m_{i,j} \beta} \in \mathbb{G}$.

where $\xi_i^{(2)} = H_{\xi^{(1)}}(\chi_i)$ and $i \in [1,n]$.

Sends $u = (\xi^{(1)}, u_1, \cdots, u_s)$ and outputs $\zeta = (\tau_1, \cdots, \tau_s)$, $\psi = (u, \chi)$ to TPA, and $\sigma = (\sigma_1, \cdots, \sigma_n)$ to CSP.

Proof (CSP,TPA): It is two-party proof protocol of retrievability between CSP (prover) and TPA (verifier) where CSP takes File as input, set of tags, a public key pk, and a set of public parameter are the common input between CSP and TP. At the end TPA return 1 or 0, where 1 shows the file is correct stored on the server. This protocol is the three move protocol between Cloud Service Provider and Third Party Auditor. Here the common input is (pk, ψ). This protocol is:

- *Commitment (CSP → TPA)*

CSP choose random $\Upsilon_j \in \mathbb{Z}_p$ *and s random* $\lambda_j \in R\, \mathbb{Z}_p$ for $j \in [1,s]$, and sends its commitment $C = (H_1', \pi)$ to TPA, where $H_1' = H_1^{\gamma}$ and $\pi \leftarrow e(\prod_{j=1}^{s} u_j^{\lambda_j}, H_2)$

- *Challenge (CSP ← TPA)*

TPA choose a random challenge set I of t indexes along with t random coefficients $v_i \in \mathbb{Z}_r$. Let Q be the set $\{(i,vi)\}_{i\in I}$ of challenge index coefficient pairs. TPA sends Q to CSP.

- *Response (CSP → TPA)*

CSP calculates the response Θ, μ as

$$\begin{cases} \sigma' & \leftarrow & \prod_{(i,v_i)\in Q} \sigma_i^{\gamma \cdot v_i}, \\ \mu_j & \leftarrow & \lambda_j + \gamma \cdot \sum_{(i,v_i)\in Q} v_i \cdot m_{i,j}, \end{cases}$$

Where, $\mu = \{\mu_j\}_{j\in[1,s]}$. *P* sends $\theta = (\sigma', \mu)$ to *V*.

- *Verification*:

TPA can check that the response was correctly formed by checking that

$$\pi \cdot e(\sigma', h) \stackrel{?}{=} e(\prod_{(i,v_i)\in Q} (\xi_i^{(2)})^{v_i}, H_1') \cdot e(\prod_{j=1}^{s} u_j^{\mu_j}, H_2).$$

The proposed audit scheme is explained above, which contains three algorithms namely key generation, tag generation and verification protocol. In key generation used is assigned a secret key sk this key is used to generate tags for files and public key pk is used to verify the integrity of stored files. In tag generation file F produces the verification parameter $\psi = (u, \chi)$. The hash value ξ can be considered as a signature of secret $\tau_1, \ldots, \tau_n$ and $u_1, \ldots, u_s$ is the encryption of this secret. Hash index table is designed according to the

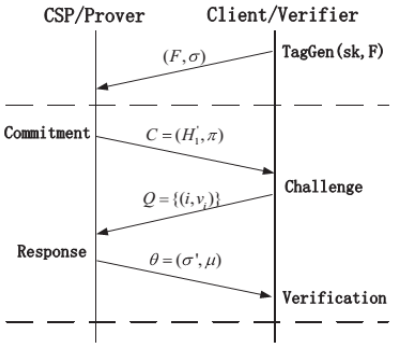

Fig. -2 Framework of interactive audit scheme

application. The index table χ is important for ensuring the security of the file. Using χ and $\xi^{(1)}$ the hash value $\xi^{(2)}$ is generated for each block of file. But it must be checked that values of ψ should be different for all processed files.

The verification process consists of three move protocol structure which contains three steps i.e. commitment, challenge and response. This is shown in Figure 2.

This protocol is similar to Schnorrs ∑ protocol. The zero knowledge property ensures that the verification process does not expose anything other than veracity of the statement in cloud. To avoid the leakage of data and tags in the verification process, the secret data {m,j} is protected by random $\lambda_j \in \mathbb{Z}_1$ .And the values $\{\lambda_j\}$ and γ are protected by simple commitment process.

4.3 Mathematical Model

It is P Class problem because: 1. Problem can be solved in polynomial time. 2. All important operations are hidden by KeyGen and TagGen algorithm. 3. The proof protocol works in polynomial time. Let S be the set of input function and output. So S = I, F, O where I is a set of Input F is a set of functions that are used to process the input provided and O be the set of output. I = I1, I2, I3 F = F1, F2, F3 O = O1, O2, O3 Input is mapped to output which is shown in following Venn diagram:

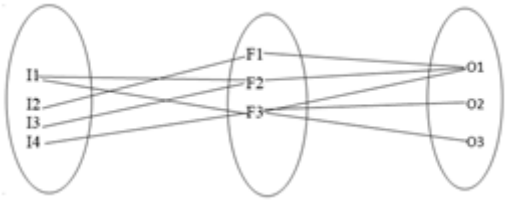

Fig-3. Venn diagram

As we have three input for the system, so we have three I as

I1: File uploaded by user.
I2: Security parameter s which is bilinear map group system.
I3: Secret key
I4: set of tags
Three functions are used F1, F2, and F3

F1: Key generation to generate the secret key.
F2: Tag Generation for file uploaded by user.
F3: Proof Technique to verify the files and perform the audit.

Also the system has three outputs O1, O2, O3

O1: File Verification
O2: File is available for download (Retrieve File after making changes by user)
O3: Perform periodic audit

5. RESULT

The implementation of system consists of implementation of KeyGen algorithm. The input given is user id and password. The bi-linear map group system is used with two groups of large prime order. The key is generated using the collision resistant hash function. While logging to the system user has to enter the key which is sent to the registered e-mail id of the user. This key is e-mailed to the user when he attempt to log in using the user id and password. The tag is generated using TagGen algorithm which accepts two inputs secret key sk and file. The file is uploaded with this tag to ensure the integrity while we are performing the audit.Audit system architecture for outsourced data in cloud.

6. CONCLUSIONS

Cloud-based outsourced storage reduces the client's burden for storage management and maintenance by providing a low cost, scalable, location-independent platform. But, the clients may not have local copy of this data. To avoid the security risks, audit services must ensure the integrity and availability of outsourced data. This dissertation presents a construction of an efficient audit service for data integrity in clouds. It also proposed an interactive audit protocol to implement the audit service based on a third party auditor.

In this audit service, the third party auditor, can issue a periodic verification to monitor the change of outsourced data by providing an optimized schedule. To realize the audit model, there is only need to maintain the security of the TPA. This technology can be easily adopted in a cloud computing environment to replace the traditional Hash-based solution.

ACKNOWLEDGEMENT

It is with the greatest pleasure and pride that I present this paper. At this moment, I cannot neglect all those who helped me in the successful completion of this paper. I am very thankful to my respected project guide Prof. P. N. Kalavadekar, Associate Professor, for his ideas and help proved to be valuable and helpful during the creation of this paper and guide me in the right path. I would also like to thank all the faculties who have cleared all the major concepts that were involved in the understanding of techniques behind this paper. Lastly, I am thankful to my friends who shared their knowledge in this field with me

REFERENCES

[1] Yan Zhu, Yan Zhu, Hongxin Huc, Gail-Joon Ahn, Stephen S. Yau. Efficient audit service outsourcing for data integrity in clouds. In: The Journal of Systems and software 85(2012) 1083-1095 2011.
[2] 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. A view of cloud computing. ACM 53 (4), 50-58, 2010.
[3] Tchifilionova, V. Security and privacy implications of cloud computing clost in the cloud. In: Camenisch, J., Kisimov, V., Dubovitskaya, M. (Eds.), Open Research Problems in Network Security. Vol. 6555 of Lecture Notesin Computer Science. Springer, Berlin/Heidelberg, pp. 149-158, 2011.
[4] Ko, R.K.L., Lee, B.S., Pearson, S. Towards achieving accountability, auditability and trust in cloud computing. In: Abraham, A., Mauri, J.L., Buford, J.F., Suzuki, J., Thampi, S.M. (Eds.), Advances in Computing and Communications. Vol. 193 of Communications in Computer and Information Science. Springer, Berlin/Heidelberg, pp. 432-444, 2011.
[5] Yavuz, A.A., Ning, P. Baf: An efficient publicly verifiable secure audit logging scheme for distributed systems. In: ACSAC, pp. 219228, 2009.
[6] Hsiao, H.-C., Lin, Y.-H., Studer, A., Studer, C., Wang, K.-H., Kikuchi, Perrig, A., Sun, H.-M., Yang, B.-Y. A study of user-friendly hash comparison schemes. In: ACSAC, pp. 105114 2009.
[7] Yumerefendi, A.R., Chase, J.S. Strong accountability for network storage. ACM Trans. Storage (TOS) 3 (3), 2007.
[8] 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), 5058.
[9] Juels Jr., A., Kaliski, B.S. Pors: proofs of retrievability for large files. In: Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, pp. 584597, 2007.
[10] Ateniese, G., Burns, R.C., Curtmola, R., Herring, J., Kissner, L., Peterson, Z.N.J., Song, D.X. Provable data possession at untrusted stores. In: Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, pp. 598609, 2007
[11] Fu, K., Kaashoek, M.F., Mazires, D. Fast and secure distributed readonly file system. ACM Trans. Comput. Syst. 20 (1), 124, 2002.

[12] Li, F., Hadjieleftheriou, M., Kollios, G., Reyzin, L. Dynamic authenticated index structures for outsourced databases. In: Chaudhuri, S., Hristidis, V., Polyzotis, N. (Eds.), SIGMOD Conference. ACM, pp. 121132, 2006

[13] Ma, D., Deng, R.H., Pang, H., Zhou, J. Authenticating query results in data publishing. In: Qing, S., Mao, W., Lopez, J., Wang, G. (Eds.), ICICS. Vol. 3783 of Lecture Notes in Computer Science. Springer, pp. 376388, 2005.

[14] Xie, M., Wang, H., Yin, J., Meng, X. 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. 782793, 2007.

[15] Ateniese, G., Pietro, R.D., Mancini, L.V., Tsudik, G. Scalable and efficient provable data possession. In: Proceedings of the 4th International Conference on Security and Privacy in Communication Networks, SecureComm, pp. 110, 2008.

[16] Erway, C.C., Kpc , A., Papamanthou, C., Tamassia, R. Dynamic provable data possession. In: Proceedings of the 2009 ACM Conference on Computer and Communications Security, CCS, pp. 213222, 2009.

[17] Shacham, H., Waters, B. Compact proofs of retrievability. In: Advances in Cryptology ASIACRY, 14th International Conference on the Theory and Application of Cryptology and Information Security, pp. 90107, 2008.

[18] Boneh, D., Franklin, M. Identity-based encryption from the weil pairing. In: Advances in Cryptology (CRYPTO2001). Vol. 2139 of LNCS, pp. 213229, 2001.