Survey on Securing Data in Cloud

Vijini Mary Kurian
Student, Karunya University,
Department of Computer Science and Engineering

Roshni Thanka.M
Lecturer, Karunya University,
Department of Computer Science and Engineering

ABSTRACT

Cloud computing is an innovative information system architecture which reduces the overall client side resource requirements. Even though the data outsourcing reduce the burden of client storage the clients no longer have physical possession of data which will leads to so many security issues. Hence the user need to make sure that their data is secure in clouds. Auditing and reporting of cloud service providers is a perfect solution that is required to be in place for all cloud services.

General Terms

Auditing, Cloud Security, Outsourced Data.

Keywords

Auditing, Cloud Computing, Cloud Service Provider, Data Integrity.

1. INTRODUCTION

Cloud computing can be simply defined as the delivery of computing and storage capacity as a service to a miscellaneous community of end recipients. With the help of cloud computing, users can easily store their data into the cloud and use on-demand high-quality applications. The concept of Cloud Computing has been resulted from the combination of Grid Computing, Utility Computing and SaaS, and essentially represents the increasing trend towards the external deployment of IT resources.

Cloud computing obtained its name cloud from the cloud symbol that is often used to represent the Internet in flow charts and diagrams. Cloud computing is accredited by virtualization technology. Virtualization technology in the sense is a host computer runs an application known as a hypervisor; this creates one or more virtual machines, which simulate physical computers so faithfully, that the simulation can be able to run any software, from any operating systems, to various end-user applications.

There are three major cloud service models [1] they are

- Infrastructure as a Service (IaaS). It allows the consumer to deploy and run arbitrary software, which include several applications and operating systems. It also provides the consumer with the capability to provision processing, storage, networks, and other fundamental computing resources.
- Platform as a Service (PaaS). It provides the consumer with the capability to deploy onto the cloud infrastructure; consumer created or acquired applications, produced using programming languages and tools supported by the provider.
- Software as a Service (SaaS). It provides the consumer with the capability to use the provider's applications running on a cloud infrastructure. The various applications are accessible from various client devices such as a web browser.

There are four major deployment models

- Private cloud. In a private cloud the cloud infrastructure is operated for a private organization. It can be managed either by the organization or by a third party, and may exist on premise or off premise.
- Community cloud. In a community cloud the cloud infrastructure is shared by several organizations and supports a specific community that has communal importance. It can be managed either by the organization or by a third party, and may exist on premise or off premise.
- Public cloud. In a public cloud the cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services.
- Hybrid cloud. The hybrid cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology, that enables data and application.

There are several complex potential security threats to virtualized computing environments [2]. For every company they have their own firewalls and anti-virus software to protect data stored on the premises. But in the case of outsourced data, control of security measures is also repudiated. Cloud computing has a number of inherent limitations and work undone that could easily breech the confidentiality, integrity and availability. However, there exist so many possibilities to have a secure and reliable cloud computing environment.

2. MODES FOR CLOUD SECURITY

There are several existing approaches which try to provide security for the data which are stored at an untrusted server.

2.1 Interactive Audit Scheme

A cryptographic interactive audit scheme also known as interactive PDP or IPDP [9]. It is used to support the audit system in clouds. Auditing is done to keep the integrity of data in cloud. This scheme is constructed on the standard model of interactive proof system, which can ensure the confidentiality of secret data and the undeceivability of invalid tags.

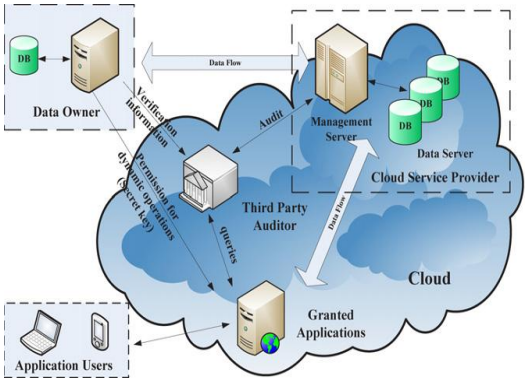

Fig.1. Architecture of interactive audit scheme

For the purpose of auditing a cryptographic interactive audit scheme *S* is used [9]. It is a collection of two algorithms and an interactive proof system, $S = (K, T, P)$:

- *K*eyGen(1s): The key generation algorithm takes a security parameter s as input, and process it to return a public-secret keypair (pk, sk);
- *T*agGen(sk, F): The tag generation algorithm takes two inputs, one is the secret key sk and other one is the file F, and returns the triples (ζ, ψ, σ) where ζ denotes the secret used to generate verification tags, ψ is the set of public verification parameters u and index information χ, i.e., $\psi = (u, \chi)$; σ represents the set of verification tags;
- *P*roof (CSP, TPA): The interactive proof system is a public two-party proof protocol of retrievability between CSP (prover) and TPA (verifier), that is $\langle \mathrm{CSP}(\mathrm{F}, \sigma), \mathrm{TPA} \rangle(\mathrm{pk}, \psi)$, where CSP takes as input a file F and a set of tags σ, and a public key pk and a set of public parameters ψ 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 correctly stored on the server and 0 means the file is corrupted.

Where, the notation P(x) denotes the subject P holds the secret x and $\langle (\mathrm{P}, \mathrm{V}) \rangle$ (x) denotes both parties P and V. It share a common data x in a protocol. This protocol is provably privacy preserving, and thus may not leak user data information to the auditor. In this audit mechanism the integrity of data is preserved by using the above scheme which keeps the original data secure from both the Cloud Service Provider as well as the Third Party Auditor. Security is assured only by sending some verification data not the full data.

2.2 Provable Data Possession

The Provable Data Possession (PDP) is one of the firs approaches which [3] helps a client that has stored data at an untrusted server to verify that the server is having the original data. The advantage is that it can be done without downloading it or storing a copy by himself. It achieves this by generating a probabilistic proof of possession by sampling random sets of blocks from the server. The client always keeps a constant amount of metadata to verify the proof.

In this technique the client have to pre-processes the file, generating a piece of metadata that will be stored locally, afterwards it transmits the file to the server, and may delete its local copy. During the pre-processing of the file, the client may edit the file to be stored at the server. The client may also increase the size of the file or include some additional metadata that is to be stored at the server. To make sure that the server has successfully stored the file, before deleting its local copy of the stored file, the client may execute a data possession challenge. It can be sending a piece of verification data. Clients may be able to encrypt a file prior to out-sourcing the storage for better security. At a later time, the client issues a challenge to the server to establish that the server has not abolished the file. The client requests that the server compute a function of the stored file, which it sends back to the client. Using the stored local metadata, the client verifies the response. During the verification the server must answer challenges from the client; any kind of failure to do so represents that the data has been compromised somewhere.

The major goal of a PDP scheme is to achieve probabilistic proof of data possession. That is to detect server misbehavior. Even when the server has deleted a fraction of the file it can be detected through this mechanism. The advantage of using this technique is to reduce the input output cost. The other advantage is that the PDP model for remote data checking supports large data sets in widely distributed storage system. Disadvantage is that it works only for static databases

2.3 Scalable and Efficient Provable Data Possession

Scalable And Efficient Provable Data Possession [4] is a highly efficient and provably secure PDP technique. It works on the concept of symmetric key cryptography, while not requiring any bulk encryption. This PDP technique efficiently supports several operations, like block modification, deletion and append. This scheme is based entirely on symmetric-key cryptography. The major idea behind it is before outsourcing the data; the data owner pre-computes a certain number of short possession verification tokens, each token covering some set of data blocks. After calculating this actual data is handed over to server. Subsequently, whenever the owner of the data wants to obtain a proof of data possession, the server must compute a short integrity check over the specified blocks corresponding to the indices and return it to owner of the data. The returned integrity check must match the corresponding value pre computed by owner of the data then the data is safe. In this scheme owner of the data has the choice of either keeping the precomputed tokens locally or outsourcing them in encrypted form to server.

The major advantage of this technique is that it supports secure and efficient dynamic operations on outsourced data blocks, including: modification, deletion and append. Supporting such operations is an important step toward practicality, since many of the application are not limited to data warehousing, that is dynamic operations need to be provided. Disadvantage is that it doesn't allows unlimited verifications and public verifiability

2.4 Dynamic Provable Data Possession

Dynamic Provable Data Possession (DPDP) [5] is one of the first efficient fully dynamic PDP solutions. It extends the PDP model to support provable updates on the stored data. This DPDP solution is based on a new variant of authenticated dictionaries, where they use rank information to organize dictionary entries. This technique make it possible to support efficient authenticated operations on files at the block level, such as authenticated insertion and deletion. The security of this construction is proved using standard assumptions.

Ateniese et al. [3] have developed another dynamic PDP solution called Scalable PDP. The idea behind it is to come up with all future challenges during setup and store pre-computed answers as metadata at the client, or at the server in an authenticated and encrypted manner. But in there technique, the number of updates and challenges a client can perform is limited and fixed a priori. Also, one cannot be able to perform block insertions anywhere only append-type insertions are possible. In this technique a client can perform any number of updates and challenges. Also the limitation of append type insertion is overcome.

2.5 POR: Proof of Retrievability

A POR [6] scheme allows a storage or back-up service or prover to produce a concise proof that a user or verifier can retrieve their target file at anytime without any obstacles. In other way it can be explained that the archive retains and reliably transmits file data sufficient for the user to recover in its entirety. A POR can be defined as a kind of cryptographic Proof of Knowledge (POK). The specialty about POR is that it is designed to manage a large file or bit string. In this protocol communication costs, number of memory accesses for the prover, and storage requirements of the user are small parameters essentially independent of the length of F. In this reference model they develop a new cryptographic building block known as a Proof of Retrievability (POR). A POR enables a user or verifier to determine that an archive or prover possesses a file or data object file without making any modification to it. More precisely, a successfully executed POR assures a verifier that the prover presents a protocol interface through which the verifier can retrieve file in its entirety. Of course, a prover can refuse to release file even after successfully participating in a POR. A POR, however, provides the strongest possible assurance of file retrievability barring changes in prover behavior.

In POR protocol the verifier stores only a single cryptographic key [6]. It is irrespective of the size and number of the files whose retrievability it seeks to verify as well as a small amount of dynamic state some tens of bits for each file. More strikingly, this scheme requires that the prover access only a small portion of a large file in the course of a POR. In fact, the portion of file touched by the prover is essentially independent of the length of file and would, in a typical parameterization, include just hundreds or thousands of data blocks. Briefly, this POR protocol encrypts file and randomly embeds a set of randomly-valued check blocks called sentinels. The use of encryption in this technique renders the sentinels indistinguishable from other file blocks. The verifier challenges the prover by specifying the positions of a collection of sentinels and he will ask the prover to return the corresponding sentinel values. If the prover has modified or deleted any substantial portion of file, then with high probability it will also have suppressed a number of sentinels. From that it is easy to find out the data loss. It is therefore tough to respond correctly to the verifier. To protect against corruption by the prover of a small portion of file, they also employ error correcting codes. This scheme is also used by the verifier to determine that an prover possesses a file or data object. Even it is having many advantages the major drawback of this POR scheme is the pre-processing or encoding of file required prior to storage with the prover. This step imposes the major disadvantage of adding some computational overhead beyond that of simple encryption or hashing as well as larger storage requirements on the prover.

2.6 Compact Proofs of Rtrievability

Shacham and Waters [7] propose protocols based on the idea of using homomorphic authenticators for file blocks, essentially block integrity values that can be efficiently aggregated to reduce bandwidth in a PoR protocol. Due to the use of integrity values for file blocks, their scheme can use a more efficient erasure code to encode the file; the block authenticators transform the erasure code into an error-correcting code. Their scheme supports an unlimited number of verifications. Compact Proofs of Retrievability is the first proof-of-retrievability schemes with full proofs of security against arbitrary adversaries in the strongest model, that of Juels and Kaliski [6]. The first scheme, built from BLS signatures and secure in the random oracle model, has the shortest query and response of any proof-of-retrievability with public verifiability. The second scheme, which builds elegantly on Pseudorandom Functions (PRFs) and is secure in the standard model, has the shortest response of any proof of retrievability scheme with private verifiability but a longer query. Both schemes rely on homomorphic properties to aggregate a proof into one small authenticator value.

The main contribution of Shacham and Waters is the construction of homomorphic linear authenticators, following a similar but informal and less efficient approach of Ateniese et al. Such authenticators allow the server to aggregate the tags of individual file blocks and authenticate a response under the improved PoR code actually, any linear functions of the blocks using a single short tag.

3. COMPARISON

(Ateniese et al. 2007) are the first one to study about the public auditability in their defined "Provable Data Possession" (PDP) model for ensuring possession of data files on untrusted storages. Their scheme make use of the the RSA-based homomorphic authenticators for auditing outsourced data. It does not sample the whole data instead of that only do sampling a few blocks of the file. But the problem with their approach is that the public auditability in their scheme needs the linear combination of sampled blocks exposed to external auditor. That means when used directly, their protocol is not fully provably privacy preserving, and thus the user data information can be leak to the auditor.

(Juels et al. 2007) describe a "Proof Of Retrievability" (PoR) model, where they used two methods called spot-checking and error-correcting codes to ensure both "possession" and "retrievability" of data files on remote storage service systems. However, the number of audit challenges a user can perform is a fixed and must be given as priori. The public auditability is also not supported in their main scheme. They describe their concept with the help of a straightforward Merkle-tree construction for public PoR but this approach only works with encrypted data. (Shacham et al. 2008) design an improved PoR scheme built from BLS signatures with full proofs of security in the security model defined in it. (Ateniese et al. 2008a) describes a partially dynamic version of the prior PDP scheme that uses only symmetric key cryptography. However, the system imposes a priori bound on the number of audits and does not support public auditability**.**

A simple comparison of the several techniques show that all above schemes provide methods for efficient auditing and provable assurance on the correctness of remotely stored data, none of them meet all the requirements for efficient audit service for data integrity in Cloud Computing, as supported in

the Interactive Audit Scheme. Table.1 and Table.2 gives the comparison of various features of various methods [9].

Table 1. Comparing computation time

Scheme	CSP computation	Client computation	Communication
PDP	O(t)	O(t)	O(1)
SPDP	O(t)	O(t)	O(t)
DPDP	O(t log n)	O(t log n)	O(t log n)
CPOR I	O(t)	O(t)	O(1)
CPOR II	O(t+s)	O(t+s)	O(s)
IPDP	O(t+s)	O(t+s)	O(s)

Table 2. Comparing other features

Scheme	Privacy	Fragment structure	Prob of detection
PDP	Yes	No	$1-(1-\rho_b)^t$
SPDP	Yes	No	$1-(1-\rho_b)^t$
DPDP	No	No	$1-(1-\rho_b)^t$
CPOR I	Yes	No	$1-(1-\rho_b)^t$
CPOR II	No	Yes	$1-(1-\rho_b)^{t.s}$
IPDP	Yes	Yes	$1-(1-\rho_b)^{t.s}$

4. CONCLUSION

The overall survey reveals that there are several technologies which are used to provide security for the data's which are stored in the clouds. Among the several technologies auditing is the efficient way which is a combination of several technologies to manage the security of outsourced data. It reduces the burden of data owners who store their data in cloud. The third party auditor will act as an intermediate and perform all the audit process without accessing the stored data. It reduces the complexity of auditing as well as provides security for data.

This paper deals with the study of various technologies which can be used to perform auditing in a better way. The future work can be concentrated on auditing with less computation time as well as communication time. Another future work is developing an audit mechanism which can be used for several types of cloud.

5. ACKNOWLEDGMENTS

I would like to express my gratitude towards my parents, my Staff members and friends who give me support an encouragement in doing this survey. I also thank for all the research scholors who did the previous study on this topic through which I got insight about my topic.

6. REFERENCES

[1] Dimitrios Zissis , Dimitrios Lekkas, Addressing cloud computing security issues, Future Generation Computer Systems, 28(2012) 583-592, December 2010.

[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, 2010, Communication ACM 53 (4), 50–58.

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

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

[5] Erway, C.C., Küpc¸ ü, A., Papamanthou, C., Tamassia, R., Dynamic provable data possession, International Proceedings of the 2009 ACM Conference on Computer and Communications Security, pp. 213–222, 2009.

[6] Juels Jr., A., Kaliski, B.S., Pors: proofs of retrievability for large files International Proceedings of the 2007 ACM Conference on Computer and Communications Security, pp. 584–597, 2007.

[7] Shacham, H., Waters, B., Compact proofs of retrievability, 14th International Conference on the Theory and Application of Cryptograpy and Information Security, pp. 90–107, 2008.

[8] Wang, C., Wang, Q., Ren, K., Lou, W., Privacy-preserving public auditing for data storage security in cloud computing, International Conference on Computer Communications Proceedings IEEE, pp. 1–9, 14-19, 2010.

[9] Yan Zhua,b,, Hongxin Huc, Gail-Joon Ahnc, Stephen S. Yauc, Efficient audit service outsourcing for data integrity in clouds, The Journal of Systems and Software 85 (2012) 1083– 1095, 2011.