Global Journal of Computer Science and Technology
Cloud & Distributed
Volume 12 Issue 12 Version 1.0 Year 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

A Survey on Secure Storage Services in Cloud Computing

By Ms. B.Tejaswi, Dr. L.V.Reddy & Ms. M.Leelavathi
Sree Vidyanikethan Engg College, Tirupati-AP, India

Abstract - Cloud computing is an emerging technology and it is purely based on internet and its environment. It provides different services to users such as Software-as-a-Service (SaaS), PaaS, IaaS, Storage-as-a-service (SaaS). Using Storage-as-a-Service, users and organizations can store their data remotely which poses new security risks towards the correctness of data in cloud. In order to achieve secure cloud storage, there exists different techniques such as flexible distributed storage integrity auditing mechanism, distributed erasure-coded data, Merkle Hash Tree(MHT) construction etc. These techniques support secure and efficient dynamic data storage in the cloud. This paper also deals with architectures for security and privacy management in the cloud storage environment.

Keywords : cloud computing, data correctness, distributed data integrity, auditing, security.

GJCST-B Classification: C.2.1

A Survey on Secure Storage Services in Cloud Computing

Ms. B.Tejaswi[α], Dr. L.V.Reddy[σ] & Ms. M.Leelavathi[ρ]

Abstract - Cloud computing is an emerging technology and it is purely based on internet and its environment. It provides different services to users such as Software-as-a-Service (SaaS), PaaS, IaaS, Storage-as-a-service (SaaS). Using Storage-as-a-Service, users and organizations can store their data remotely which poses new security risks towards the correctness of data in cloud. In order to achieve secure cloud storage, there exists different techniques such as flexible distributed storage integrity auditing mechanism, distributed erasure-coded data, Merkle Hash Tree(MHT) construction etc. These techniques support secure and efficient dynamic data storage in the cloud. This paper also deals with architectures for security and privacy management in the cloud storage environment.

Keywords : cloud computing, data correctness, distributed data integrity, auditing, security.

I. Introduction

From the perspective of data security in cloud, this has always been an important aspect of quality of service. The data stored in the cloud may be frequently updated by the users, including insertion, deletion, modification, appending, reordering, etc. To ensure storage correctness under dynamic data update is hence of paramount importance. In [1], as cloud-based services continues to grow, it has become clear that one of the key barriers to rapid adoption of enterprise cloud services is customer concern over data security (confidentiality, integrity, and availability). According to sun micro systems, the concept of transparent security makes the case that the intelligent disclosure of security design, practices, and procedures can help to improve customer confidence. According to K Ren, C.Wang and Q.Wang (Ref [2]), cloud storage allows data owners to outsource their data to cloud. However owners no longer have physical possession of the outsourced data raises big security concerns on the storage correctness. Hence, enabling secure storage auditing in the cloud environment with new approaches becomes imperative and challenging. In [3], the authors specified a new scheme called Proof of Retrievability (POR), is a kind of cryptographic proof and is designed to handle large file. In [4], the authors introduced a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The PDP model for remote data checking supports large data sets in widely-distributed storage systems.In [5], the authors specified third party auditing which is important in creating online service-oriented-storage economy. It allows customers to evaluate risks. Also the authors introduced two types of auditing mechanisms; those are internal auditing and external auditing. In [6], the authors introduced privacy-preserving protocols. To make storage services accountable for data loss, authors presented protocols that allow a third- party auditor to periodically verify the data stored by a service and assist in returning the data intact to the customer. Most importantly, protocols for privacy-preserving never reveal the data contents to the auditor. In [7], the authors dealt with data dynamics. In order to achieve efficient data dynamics authors improved the Proof of Retrievability model by manipulating the classic Merkle Hash Tree (MHT) construction for block tag authentication. In [8], the authors (Q. Wang, K. Ren, W. Lou, and Y. Zhang) proposed a novel dependable and secure data storage scheme with dynamic integrity assurance. Based on the principle of secret sharing and erasure coding, they first proposed a hybrid share generation and distribution scheme to achieve reliable and fault-tolerant initial data storage by providing redundancy for original data components. In [10], the authors introduced the concept of an aggregate signature, and presented security models for such digital signatures, and given several applications for aggregate signatures. They have constructed an e-client aggregate signature from a recent short signature scheme based on bilinear maps. In [10], the authors described a privacy manager for cloud computing, which reduces the risk to the cloud computing user of their private data being stolen or misused, and also assists the cloud computing provider to conform to privacy law. They described different possible architectures for privacy management in cloud computing; given an algebraic description of obfuscation, one of the features of the privacy manager; and described how the privacy manager might be used to protect private metadata of online photos. In [11], the authors explored a newly emerging problem of information leakage caused by indexing in the cloud.

Author α : M.Tech Student, Department of CSE, Sree Vidyanikethan Engg College, Tirupati-517102, AP, India. E-mail : tej.519@gmail.com
Author σ : Professor, Department of CSE, Sree Vidyanikethan Engg College, Tirupati-517102, AP, India. E-mail : lakkireddy.v@gmail.com
Author ρ : M.Tech Student, Department of CSE, Sree Vidyanikethan Engg College, Tirupati-517102,AP, India.
E-mail : leela9.mullangi@gmail.com

They designed a three-tier data protection architecture to accommodate various levels of privacy concerns by users. According to the architecture, they developed a novel portable data binding technique to ensure strong enforcement of users' privacy requirements at server side.

II. Methods

a) *Concept of Transparent security*

According to NIST (National Institute of Standards and Technology)Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

i. *Definition of Transparent security*

Transparent security can be defined as appropriate disclosure of the governance aspects of security design, policies, and practices. A security policy explains the high level approach to security and typically represents an organization’s executive management position on security and risk. The policy might state that user data will be protected from unauthorized access both while being stored in the cloud and while in transit. The security design might then go into more detail by specifying that file-level encryption will be used along with an identity management implementation that restricts access to stored data. Security practices might also drill down further, describing the processes for proper management of encryption keys. The actual implementation would then choose a specific encryption algorithm such as Advanced Encryption Standard (AES).

ii. *Transparent Security Principles*

The following transparent security principles help to identify the types of information that should and/or should not be disclosed. The following conditions are examples of when disclosure is recommended:

- Principle-1—Disclosure of common security policies and practices. Common security features such as the use of firewalls and encryption of data in transmission or at rest should be disclosed because they are considered basic security features that most security people would expect to be in place anyway.
- Principle-2—Disclosure when mandated. When disclosure is imperative due to a legal or regulatory requirement, then this disclosure must be performed.
- Principle-3—Security architecture. Security architectural details that may either help or hinder security management should be disclosed. For example, the implementation of secure by default configuration should be disclosed. However, if these types of details also create a security risk as described in other items below, disclosure would not be appropriate.
- Principle-4—Governance. It means responsibilities of the customer versus those of the cloud providers, should be clearly articulated so that the customers are clear on what they must do themselves to protect their data.

b) *Proof of Retrievability (POR) Model*

POR protocol encrypts F and randomly embeds a set of randomly-valued check blocks called *sentinels*. The use of encryption here renders the sentinels indistinguishable from other file blocks. The verifier challenges the prover by specifying the positions of a collection of sentinels and asking the prover to return the associated sentinel values. 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. It is therefore, unlikely to respond correctly to the verifier. To protect against corruption by the prover of a small portion of F error correcting codes are there. Let F’ refer to the full, encoded file stored with the prover.

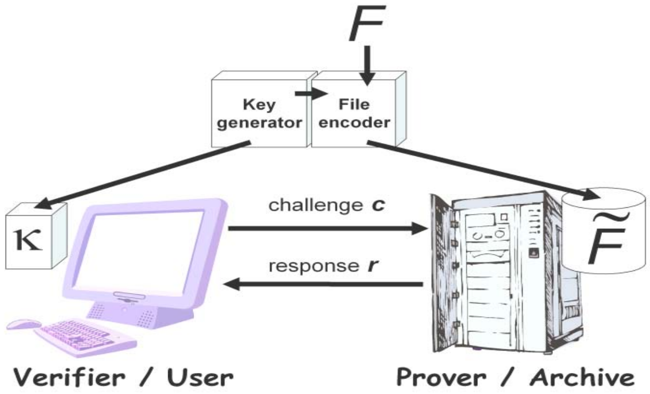


Fig. 2.2.1 : Schematic diagram of a POR system

An encoding algorithm transforms a raw file F into an encoded file *F* to be stored with the prover / archive. A key generation algorithm produces a key stored by the verifier and used in encoding.

c) *Provable Data Possession (PDP) Model*

Provable data possession (PDP) model provides probabilistic proof that a third party stores a file. The model is unique in that it allows the server to access small portions of the file in generating the proof; the above technique is used to access the entire file. Within this model, the authors (Ref [4]) give the first provably-secure scheme for remote data checking. The client stores a small O (1) amount of metadata to verify the server’s proof. This model uses homomorphic verifiable tags. Because of the homomorphic property, tags computed for multiple file blocks can be combined into a single value. The client pre-computes tags for each block of a file and then stores the file and its tags with a server. At a later time, the client can verify that the server possesses the file by generating a random challenge against a randomly selected set of file blocks. Using the queried blocks and their corresponding tags, the server generates a proof of possession. The client is thus convinced of data possession, without actually having to retrieve file blocks. A PDP protocol checks that an outsourced storage site retains a file, which consists of a collection of n blocks. The client C (data owner) pre-processes the file, generating a piece of metadata that is stored locally, transmits the file to the server S, and may delete its local copy. The server stores the file and responds to challenges issued by the client.

d) Third Party Auditing (TPA)

Third Party Auditor: 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. The task of TPA is to verify integrity of the dynamic data stored in the cloud. Users rely on the CS for cloud data storage and maintenance. They may also dynamically interact with the CS to access and update their stored data for various application purposes. The users may resort to TPA for ensuring the storage security of their outsourced data, while hoping to keep their data private from TPA. We consider the existence of a semi-trusted CS as does. Namely, in most of time it behaves properly and does not deviate from the prescribed protocol execution. However, during providing the cloud data storage based services, for their own benefits the CS might neglect to keep or deliberately delete rarely accessed data files which belong to ordinary cloud users. Moreover, the CS may decide to hide the data corruptions caused by server hacks or Byzantine failures to maintain reputation. We assume the TPA, who is in the business of auditing, is reliable and independent, and thus has no incentive to collude with either the CS or the users during the auditing process.

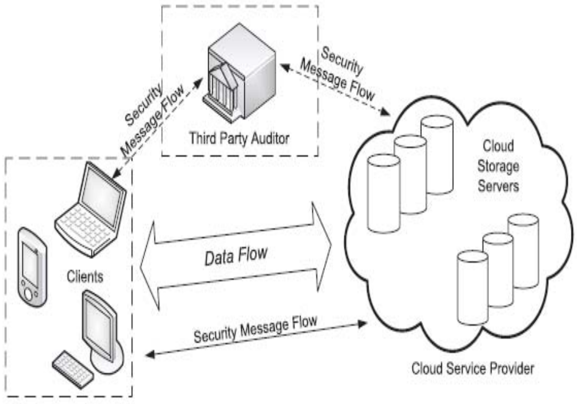


Fig. 2.4.1 : Cloud data storage architecture

e) Secret sharing and Erasure coding

Secret Sharing- Shamir proposed an (*m, n*) Secret Sharing (SS) scheme based on polynomial interpolation, in which *m* of *n* shares of a secret are required to reconstruct the secret.

Erasure Code -An (*k, n*) erasure code encodes a block of data into *n* fragments, each has 1/*k* the size of the original block and any *k* fragments can be used to reconstruct the original data block. Examples are Reed Solomon (RS) codes and Rabin's Information Dispersal Algorithm.

In the basic scheme, suppose a sensor node *v* has *data* to be stored locally. To protect *data*, it can perform the following operations to ensure the data integrity and confidentiality:

- **Step 1**: Generate a random session key *kr* and compute the keyed hash value *h*(*data, kr*) of *data*.
- **Step 2:** Encrypt *data*, *h* (*data, kr*) with *kr* and obtain *{data, h (data, kr) }kr* .
- **Step 3**: Encrypt *kr* using the key KUV shared between the authorized users and itself. This key can be either symmetric or asymmetric depending on the chosen user access control mechanism, which is independent to our design here and will not be discussed in this paper.
- **Step 4:** Store DATA $=< \{data, h(data, kr)\}kr , \{kr\}KUV >$ and destroy *kr*.

III. Architectures

a) Privacy Manager in the Client

Privacy Manager Software on the client helps users to protect their privacy when accessing cloud services. A central feature of the Privacy Manager is that it can provide obfuscation and de-obfuscation service, to reduce the amount of sensitive information held within the cloud. Privacy Manager allows the user to express privacy preferences about the treatment of their personal information, including the degree and type of obfuscation used. Personae – in the form of icons that correspond to sets of privacy preferences – can be used to simplify this process and make it more intuitive to the user.

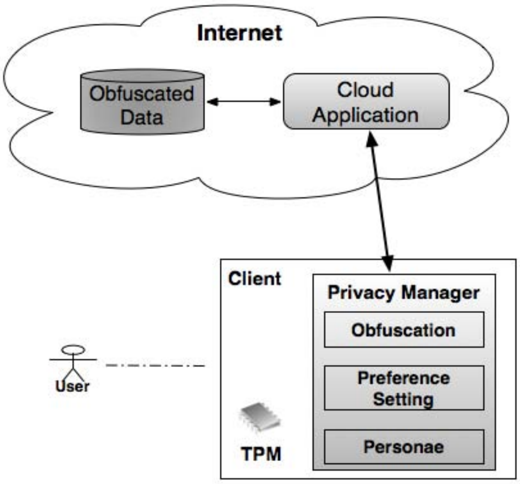
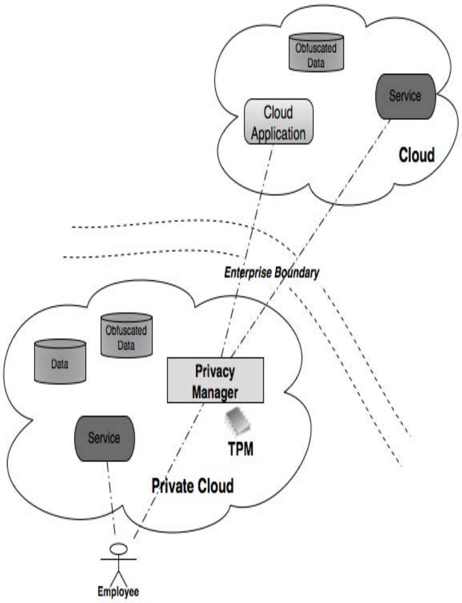


Fig. 3.1.1 : Client-Based Privacy Manager

b) Privacy Manager in a Hybrid Cloud

Privacy Manager may be deployed in a local network, or a private cloud, to protect information relating to multiple parties. This would be suitable in environments, such as enterprise environments, where local protection of information is controlled in an adequate manner and its principal use would be to control personal information passing to a public cloud.

Fig. 3.2.1 : Enterprise-focused Privacy Manager



Advantages to this approach include that the benefits of the cloud can be reaped within the private cloud, including the most efficient provision of the Privacy Manager functionality. It can provide enterprise control over dissemination of sensitive information, and local compliance. The Privacy Manager would act on behalf of the user and decide the degree of data transfer allowed, based upon transferred user policies and the service context, and preferably also an assessment of the trustworthiness of the service provision environment.

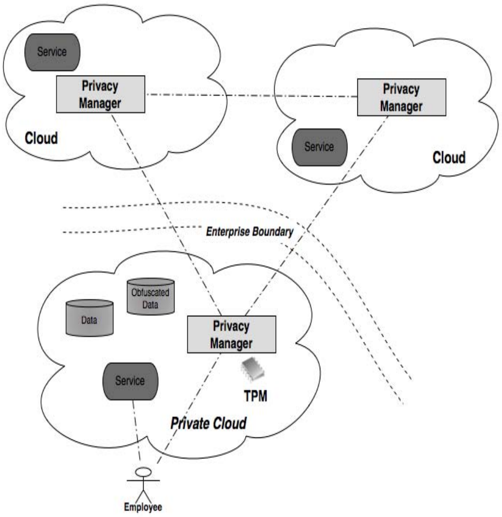


Fig. 3.2.2 : Privacy Manager within the Cloud

IV. Conclusion

This paper dealt with different security models to protect the data which is stored in the cloud. According to the user requirements, they may choose the most appropriate model. However, in the case of Third Party Auditing (TPA), cloud data storage security is critical because of its poor service quality. This paper also dealt with different architectural representations for privacy management. We provide the extension of the proposed one to support TPA, so that the users can safely delegate the integrity checking tasks.

References Références Referencias

1. Sun Microsystems, Inc., "Building Customer Trust in Cloud Computing with Transparent Security," https://www.sun.com/ offers/details/sun_ transparency.xml, White paper, Nov. 2009
2. K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.
3. A. Juels and B.S. Kaliski Jr., "PORs: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, Oct. 2007.
4. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 598-609, Oct. 2007.
5. M.A. Shah, M. Baker, J.C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest," Proc. 11th USENIX Workshop Hot Topics in Operating Systems (HotOS '07), pp. 1-6, 2007.

6. M.A. Shah, R. Swaminathan, and M. Baker, “Privacy- Preserving Audit and Extraction of Digital Contents,” Cryptology ePrint Archive, Report 2008/186, http://eprint.iacr.org, 2008.
7. Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing,” Proc. 14th European Conf. Research in Computer Security (ESORICS’09), pp. 355-370, 2009.
8. Q. Wang, K. Ren, W. Lou, and Y. Zhang, “Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance,” Proc. IEEE INFOCOM, pp. 954-962, Apr.2009.
9. D. Boneh, C. Gentry, B. Lynn, and H. Shacham, “Aggregate and Verifiably Encrypted Signatures from Bilinear Maps,” Proc. 22nd Int’l Conf. Theory and Applications of Cryptographic techniques (Eurocrypt ’03), pp. 416-432, 2003.
10. S. Pearson, Y. Shen, and M. Mowbray, “A Privacy Manager for Cloud Computing,” Proc. Int’l Conf. Cloud Computing (CloudCom), pp. 90-106, 2009.
11. A. Squicciarini, S. Sundareswaran, and D. Lin, “Preventing Information Leakage from Indexing in the Cloud,” Proc. IEEE Int’l Conf. Cloud Computing, 2010.