International Journal of Current Research and Modern Education (IJCRME)
ISSN (Online): 2455 - 5428
(www.rdmodernresearch.com) Volume I, Issue I, 2016

SECURE DATA TRANSMISSION IN CLOUD COMPUTING

J. Sasi Devi*, R. Aarthy, M. S. Dhageer Basha*** & A. Daniel Gnansekar*****

* Associate Professor & Head, Department of Computer Science & Engineering, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu
** Assistant Professor, Department of Computer Science & Engineering, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu
*** UG Scholar, Department of Mechanical Engineering, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu

Abstract:

Storage-as-a-Service offered by cloud service providers (CSPs) is a paid facility that enables organizations to outsource their sensitive data to be stored on remote servers. In this paper, we propose a cloud-based storage scheme that allows the data owner to benefit from the facilities offered by the CSP and enables indirect mutual trust between them. The proposed scheme has four important features: (i) it allows the owner to outsource sensitive data to a CSP, and perform full block-level dynamic operations on the outsourced data, i.e., block modification, insertion, deletion, and append, (ii) it ensures that authorized users (i.e., those who have the right to access the owner's file) receive the latest version of the outsourced data, (iii) it enables indirect mutual trust between the owner and the CSP, and (iv) it allows the owner to grant or revoke access to the outsourced data. We discuss the security issues of the proposed scheme. Besides, we justify its performance through theoretical analysis and a prototype implementation on Amazon cloud platform to evaluate storage, communication, and computation overheads.

Introduction to Cloud Computing:

Cloud computing is a colloquial expression used to describe a variety of different types of computing concepts that involve a large number of computers connected through a real-time communication network (typically the internet). Cloud computing is a jargon term without a commonly accepted non-ambiguous scientific or technical definition. in science, cloud computing is a synonym for distributed computing over a network and means the ability to run a program on many connected computers at the same time. the phrase is also, more commonly used to refer to network based services which appear to be provided by real server hardware, which in fact are served up by virtual hardware, simulated by software running on one or more real machines. Such virtual servers do not physically exist and can therefore be moved around and scaled up (or down) on the fly without affecting the end user - arguably, rather like a cloud.

Advantages:

The cloud also focuses on maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but as dynamically re-allocated per demand. This can work for allocating resources to users in different time zones. For example, cloud computer facilities, which serve European users during European business hours with a specific application while the same resources are getting reallocated. And serve North American users during North America's business hours with another application.

Characteristics: On-Demand Self-Service:

A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

Broad Network Access:

Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms.

Resource Pooling:

The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

Rapid Elasticity:

Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear unlimited and can be appropriated in any quantity at any time.

Measured Service:

Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service. Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Related Works:

Storage outsourcing is a rising trend which prompts a number of interesting security issues, many of which have been extensively investigated in the past. However, Provable Data Possession (PDP) is a topic that has only recently appeared in the research literature. The main issue is how to frequently, efficiently and securely verify that a storage server is faithfully storing its client's (potentially very large) outsourced data. The storage server is assumed to be untrusted in terms of both security and reliability. (In other words, it might maliciously or accidentally erase hosted data; it might also relegate it to slow or off-line storage.) The problem is exacerbated by the client being a small computing device with limited resources. Prior work has addressed this problem using either public key cryptography or requiring the client to outsource its data in encrypted form. In this paper, we construct a highly efficient and provably secure PDP technique based entirely on symmetric key cryptography, while not requiring any bulk encryption. Also, in contrast with its predecessors, our PDP technique allows outsourcing of dynamic data, i.e, it efficiently supports operations, such as block modification, deletion and append.

Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise. It moves the application software and databases to the centralized large data centers, where the management of the data and services may not be fully trustworthy. This unique paradigm brings about many new security challenges, which have not been well understood. This work studies the problem of ensuring the integrity of data storage in Cloud Computing. In particular, we consider the task of allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The introduction of TPA eliminates the involvement of client through the auditing of whether his data stored in the cloud is indeed intact, which can be important in achieving economies of scale for Cloud Computing. The support for data dynamics via the most general forms of data operation, such as block modification, insertion and deletion, is also a significant step toward practicality, since services in Cloud Computing are not limited to archive or backup data only. While prior works on ensuring remote data integrity often lacks the support of either public verifiability or dynamic data operations, this paper achieves both. We first identify the difficulties and potential security problems of direct extensions with fully dynamic data

updates from prior works and then show how to construct an elegant verification scheme for seamless integration of these two salient features in our protocol design. In particular, to achieve efficient data dynamics, we improve the Proof of Retrievability model by manipulating the classic Merkle Hash Tree (MHT) construction for block tag authentication. Extensive security and performance analysis show that the proposed scheme is highly efficient and provably secure.

Commonly, traditional access control techniques assume the existence of the data owner and the storage servers in the same trust domain. This assumption, however, no longer holds when the data is outsourced to a remote CSP, which takes the full charge of the outsourced data management, and resides outside the trust domain of the data owner. CSP needs to be safeguarded from a dishonest owner, who attempts to get illegal compensations by falsely claiming data corruption over cloud servers. This concern, if not properly handled, can cause the CSP to go out of business.

Proposed Work:

In the proposed system we design and implementation of a cloud-based storage scheme that has the following features: (i) It allows a data owner to outsource the data to a CSP, and perform full dynamic operations at the block-level, i.e., it supports operations such as block modification, insertion, deletion, and append; (ii) It ensures the newness property, i.e., the authorized users receive the most recent version of the outsourced data; (iii) It establishes *indirect* mutual trust between the data owner and the CSP since each party resides in a different trust domain; and (iv) It enforces the access control for the outsourced data.

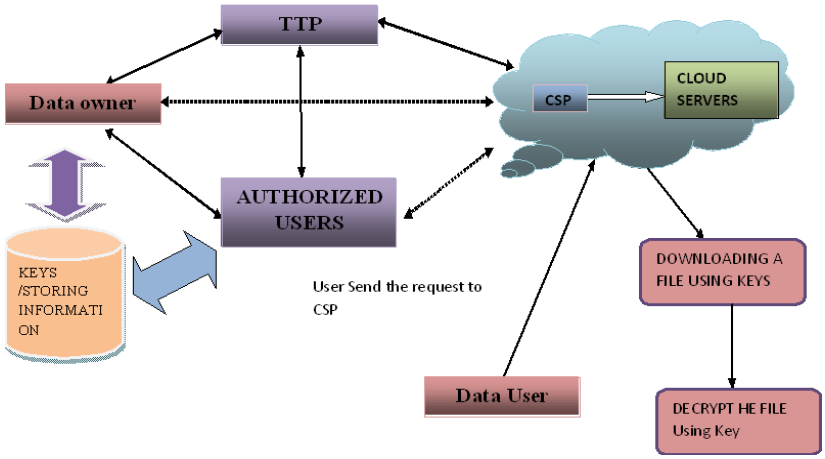

Figure 1: Proposed Architecture

System Models:

The system model includes the following.

- File encryption
- File upload to Service Providers
- Dynamic Operations on the Outsourced Data
- Data Access and Cheating Detection
- File decryption

File Encryption:

The first module in this project is file encryption module. This module is designed for encrypt the file before outsourcing the file into cloud service providers. The encryption process done by the dynamic data owner to prevent their data from the

unauthorized users. During the encryption time the secret key for the file to decrypt the file is produced. The owners have to keep the secret key. When they are retrieving the data from the cloud service providers the data will be in encrypted form. So this module plays an important role in our project.

File Upload to Service Providers:

The data owner can not directly upload their files into the cloud service providers. The data owner first has to upload their files into the Trusted Third Party. The TTP in our project is a trusted intermediate between the cloud service providers and the data owner. The TTP first receives the data from the data owner and forward the file to the cloud service providers, when the file is receives at cloud service providers from the TTP then it sends a confirmation mail that the file is uploaded at the cloud service providers to the data owner.

Dynamic Operations on the Outsourced Data:

The data owner can modify their file after uploading their file into the cloud service providers. They can do the operations dynamically on the data. So the authorized users can access recently updated version of the outsourced data. Only the data owner can change the data dynamically. The data can be deleted, updated or edited by the data owner.

Data Access and Cheating Detection:

An authorized user sends a data-access request to both the CSP and the TTP to access the outsourced file. The outsourced data can be only retrieved by the authorized users. The TTP has to check whether the users are authorized persons or not. To check the authorization the CSP and the TTP check the secret key of the particular file which has the data request by the users. If the secret keys match with the database then only they can download the file and decrypt it. if there any unauthorized users try to access the data the notification will send to the TTP.

File Decryption:

The last module in this project is file decryption. In this module the encrypted file will return back into its original form. For the decryption process the algorithm need the key which created at the time of encryption. The data owner keeps the key generated at encryption process. After enter the key the algorithm will decrypts the file and returns the data in a readable manner which can be understand by the users.

Algorithms:

Identity Based Encryption:

ID-based encryption (or identity-based encryption (IBE)) is an important primitive of ID-based cryptography. As such it is a type of public-key encryption in which the public key of a user is some unique information about the identity of the user (e.g. a user's email address). This can use the text-value of the name or domain name as a key or the physical IP address it translates to. The first implementation of an email-address based PKI, which allowed users to verify digital signatures using only public information such as the user's identifier.

ID-based encryption was proposed, however only able to give an instantiation of identity-based signatures. Identity-based encryption remained an open problem for many years. One example of the research leading up to identity-based encryption. Identity-based systems allow any party to generate a public key from a known identity value such as an ASCII string. A trusted third party, called the private key generator (PKG), generates the corresponding private keys. To operate, the PKG first publishes a master public key, and retains the corresponding master private key (referred to as master key). Given the master public key, any party can compute a public key

corresponding to the identity ID by combining the master public key with the identity value. To obtain a corresponding private key, the party authorized to use the identity ID contacts the PKG, which uses the master private key to generate the private key for identity ID.

Identity-based systems allow any party to generate a public key from a known identity value such as an ASCII string. A trusted third party, called the Private Key Generator (PKG), generates the corresponding private keys. To operate, the PKG first publishes a master public key, and retains the corresponding master private key (referred to as master key). Given the master public key, any party can compute a public key corresponding to the identity ID by combining the master public key with the identity value. To obtain a corresponding private key, the party authorized to use the identity ID contacts the PKG, which uses the master private key to generate the private key for identity ID.

As a result, parties may encrypt messages (or verify signatures) with no prior distribution of keys between individual participants. This is extremely useful in cases where pre-distribution of authenticated keys is inconvenient or infeasible due to technical restraints. However, to decrypt or sign messages, the authorized user must obtain the appropriate private key from the PKG. A caveat of this approach is that the PKG must be highly trusted, as it is capable of generating any user's private key and may therefore decrypt (or sign) messages without authorization.

Conclusion:

This project proposed a scheme related to outsourcing the storage of data. The owner is capable of not only archiving and accessing the data stored by the csp, but also updating and scaling this data on the remote servers. The remotely stored data can be not only accessed by authorized users, but also updated and scaled by the owner. After updating, authorized users should receive the latest version of the data. A TTP is able to determine the dishonest party. The outsourced data is protected using the broadcast encryption and decryption algorithm.

References:

1. C. Erway, A. K¨upc¸ ¨ u, C. Papamanthou, and R. Tamassia, (2009)"Dynamic provable data possession," in Proceedings of the 16th ACM Conference on Computer and Communications Security, pp. 213–222.
2. Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, (2009)"Enabling public verifiability and data dynamics for storage security in cloud computing," in Proceedings of the 14th European Conference on Research in Computer Security, pp. 355–370.
3. A. F. Barsoum and M. A. Hasan, (2010)"Provable possession and replication of data over cloud servers," Centre For Applied Cryptographic Research,Report2010/32,http://www.cacr.math..ca/techreports/2010/cacr2010-32.pdf.
4. R. Curtmola, O. Khan, R. Burns, and G. Ateniese,(2008) "MR-PDP: multiple-replica provable data possession," in 28th IEEE ICDCS, pp. 411–420.
5. A. F. Barsoum and M. A. Hasan,(2011) "On verifying dynamic multiple data copies over cloud servers," Cryptology ePrint Archive, Report 2011/447, http://eprint.iacr.org/.
6. K. D. Bowers, A. Juels, and A. Oprea, (2009)"HAIL: a high-availability and integrity layer for cloud storage," in CCS '09: Proceedings of the 16th ACM conference on Computer and communications security. New York, NY, USA: ACM, pp. 187–198.

7. Y. Dodis, S. Vadhan, and D. Wichs, (2009) "Proofs of retrievability via hardness amplification," in Proceedings of the 6th Theory of Cryptography Conference on Theory of Cryptography.
8. A. Juels and B. S. Kaliski,(2007) "PORs: Proofs of Retrievability for large files," in CCS'07: Proceedings of the 14th ACM conference on Computer and communications security. ACM, pp. 584–597.
9. H. Shacham and B. Waters,(2008) "Compact proofs of retrievability," in ASIACRYPT '08, pp. 90–107.
10. M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, (2003)" Plutus: Scalable secure file sharing on untrusted storage," in Proceedings of the FAST 03: File and Storage Technologies.