International Journal of Future Generation Communication and Networking
Vol. 15, No. 1, (2022), pp. 181–187

A Cloud Secure Storage Mechanism Based On Data Dispersion And Encryption

S. Vishnu Priya [1], Dr. C. Siva Balaji Yadhav[2], P. Ramesh[3]

[1]PG Scholar, Dept. of CSE, VEMU Institute of Technology, P.Kothakota, Chittoor.

[2]Associate Professor, Dept. of CSE, VEMU Institute of Technology, P.Kothakota, Chittoor.

[2]Assistant Professor, Dept. of CSE, VEMU Institute of Technology, P.Kothakota, Chittoor.

svishnupriya@gmail.com[1] sivabalaji2233@gmail.com[2] rameshperamalasetty@gmail.com[3]

Abstract

Cloud storage service has shown its great power and wide popularity which provides fundamental support for rapid development of cloud computing. However, due to management negligence and malicious attack, there still lie enormous security incidents that lead to quantities of sensitive data leakage at cloud storage layer. From the perspective of protecting cloud data confidentiality, this paper proposed a Cloud Secure Storage Mechanism named CSSM. To avoid data breach at the storage layer, CSSM integrated data dispersion and distributed storage to realize encrypted, chucked and distributed storage. In addition, CSSM adopted a hierarchical management approach and combined user password with secret sharing to prevent cryptographic materials leakage. The experimental results indicate that proposed mechanism is not only suitable for ensuring the data security at storage layer from leakage, but also can store huge amount of cloud data effectively without imposing too much time overhead. For example, when users upload/download 5G sized file with CSSM, it only takes

646seconds/269seconds, which is acceptable for users.

Keywords: Exploratory Data Analysis, Cloud computing, data dispersion, data encryption, key management, storage security.

1. Introduction

Cloud computing has shown remarkable development in recent decades. When the storage as a service, it occupies the center stage and backbone for many applications, such as pattern recognition image forensic and forgery detection. As a result, larger volumes of data will be a part of the cloud area. In the cloud industry, Amazon Web Service (AWS) has become the de facto standard. As the core component of the Opens tack that follows this standard, Swift has become one of the most popular cloud storage mechanism. However, Opens tack Swift mechanism still faces many real security threats while providing convenient services. According to Cloud Security Alliance's top threat case analysis report released in 2018, two thirds of the cases will cause user data leakage, mainly due to management negligence and malicious attacks. For instance, under default configuration, OpenStack Swift mechanism typically stores data in plaintext for the sake of performance.

That will lead unauthorized access to user data at storage layer. In addition, Security Report released by Openstack Vulnerability Management Team VMT, the Swift mechanism may leak user data or configuration information in virtue of security vulnerabilities.

ISSN: 2233-7857IJFGCN

2. Literature Review

Shah et al proposed a cloud-oriented data security storage mechanism under the framework of Apache Spark, which prevents data leakage and improves the security of Apache Spark framework. To protect user data on the cloud, different encryption schemes have been adopted to avoid information leakage during machine learning process. Nevertheless, above researches require secure key management mechanisms to prevent cryptographic materials exposure.

Zerfos et al.constructed a secure distributed storage system based on Hadoop system, which keep the confidentiality of cloud data through data dispersion and encryption. It performs the data decryption and assembly tasks before reading data. To prevent the keys from being stolen, this method requires key cache server and all keys should be stored in memory only. Some approaches introduced independent third party to manage the key. It is assumed that third parties stay trusted. However, the assumption cloud not always exists in the real cloud storage environments. Wang et al. presented a data privacy preserving scheme for sensor-cloud system, based on edge computing and differential storage method. In this scheme, user data would be divided into different parts and stored in local, edge and cloud layer respectively. But the scheme relies on the characteristics of data from wireless sensor networks, and requires skilled users to manage the edge servers.

Zheng et al. provided a cloud data duplication scheme to detect and remove identical user data in the cloud. However, from the perspective of preventing data loss due to disaster, a certain number of copies should be sent to multiple regions. In a word, to protect cloud data from leakage at storage layer, this paper presents CSSM, a Cloud Secure Storage Mechanism. CSSM combines data dispersion with data encryption, so that large-scale cloud data and keys would be stored in chunked cipher texts. On this basis, user password and secret sharing are introduced to further protect keys security.

Data sharing becomes an exceptionally attractive service supplied by cloud computing platforms because of its convenience and economy. As a potential technique for realizing finegrained data sharing, attribute-based encryption (ABE) has drawn wide attentions. However,most of the existing ABE solutions suffer from the disadvantages of high computation overhead and weak data security, which has severely impeded resource-constrained mobile devices to customize the service. The problem of simultaneously achieving fine-grainedness, high efficiency on the data owner’s side, and standard data confidentiality of cloud data sharing actually still remains unresolved. The proposed scheme eliminates a majority of the computation task by adding system public parameters besides moving partial encryption computation offline. In addition, a public ciphertext test phase is performed before the decryption phase, which eliminates most of computation overhead due to illegitimate ciphertexts. For the sake of data security, a Chameleon hash function is used to generate an immediate cipher text, which will be blinded by the offline ciphertexts to obtain the final online ciphertexts. The proposed scheme is proven secure against adaptively chosen-ciphertext attacks, which is widely recognized as a standard security notion. Extensive performance analysis indicates that the proposed scheme is secure and efficient.

ISSN: 2233-7857IJFGCN

3. Methodology

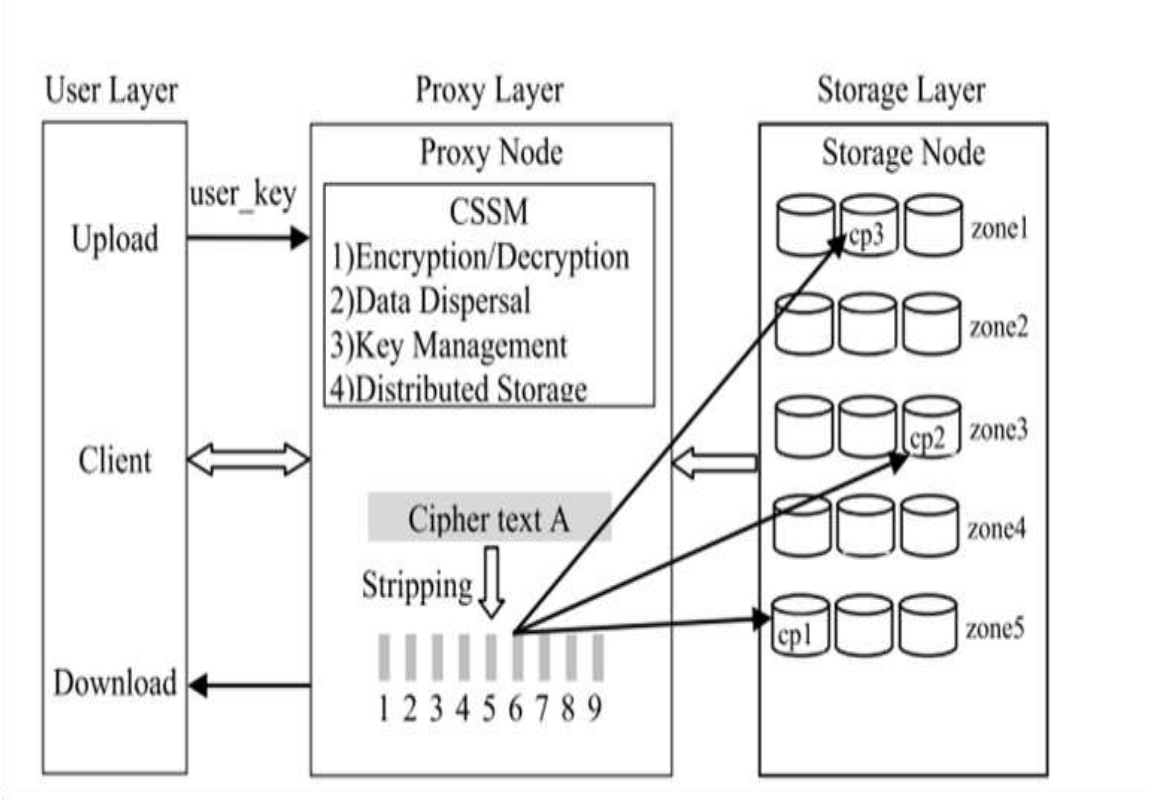

Figure: System Architecture

3.1 CSSM (Cloud Secure Storage Mechanism)

The proposed mechanism is to secure cloud storage against data breach, which may be the result of targeted attack (e.g. disk cloning) or management negligence (e.g. misconfiguration), in case hackers even some malicious administrator is able to steal user data. Aiming at this goal, data dispersion or encryption is the most commonly adopted way in numerous cases. Both techniques could provide privacy-preserving, but they also come with inherent risks. Data dispersion spreads data pieces across different storage areas, but there still lies an opportunity to recover data when attackers obtain enough pieces. Data encryption technology stores data in cipher texts by encrypting data with cryptographic keys. However, attackers can still recover the original data by stealing the keys. That raises the problem of key protection and management. Therefore, to maximize the confidentiality of cloud data storage, the proposed mechanism should make full use of the advantages of the method and effectively control its disadvantages. Meanwhile, the increased cost of the mechanism should be within a reasonable range. Specifically, following properties should be met:

Property 1: From the perspective of protecting cloud data confidentiality, any user data stored in the cloud would not be released, viewed, stolen or used by unauthorized individual, such as hackers or some malicious administrator.

Property 2: On the basis of property 1, any parameters like cryptographic keys, which related to keep cloud data confidential, should also be protected.

Property 3: The additional performance overhead of deploying proposed mechanism should be within the user's acceptance.

ISSN: 2233-7857IJFGCN

3.2 Architecture Overview

To realize primary object and properties above, this paper presents CSSM, a cloud secure storage mechanism. As shown in Figure 1, CSSM could be divided into three layers: The user layer, the proxy layer, and the storage layer. Specifically, the main functions of each layer are as follows:

1) User Layer: This layer is deployed on the user's machine, and the user operates (upload, download, etc.) cloud data through the client.

2) Proxy Layer: This layer is deployed in the cloud and composed of proxy nodes with trusted execution environments, such as Intel SGX technology and ARM Trust Zone technology. In trusted execution environment, CSSM programs could perform as expected. CSSM in proxy layer includes four modules: data encryption/decryption, data dispersal, key management and distributed storage.

- **Encryption/Decryption:** This module is used to encrypt user uploaded data and decrypt user downloaded data.
- **Data Dispersal:** According to the data dispersal model, the cipher texts is divided into several small blocks.
- **Key Management:** This module is not only responsible for the generation and maintenance of the key, but also uses the hierarchical key management approach to protect the key.
- **Distributed storage:** This module distributes chunked and encrypted data to storage layer.

3) Storage Layer: This layer consists of a number of storage nodes that are used to store chunked and encrypted data. Considering data loss or unavailability caused by accident like equipment damage or natural disasters, cloud service providers divide large number of storage nodes into several zones, each of which acts as a failure boundary between multiple copies of the same data.

4．Implementation

4.1 Proxy

Client Registration

Proxy has to login with valid credentials into the system. After successful login proxy is responsible to register sender and receiver information and transfers login details to receiver/sender through mail.

View Requests

Proxy can view the request for files from the receivers and sends the trapdoor which decrypt the encrypted information using mail.

Logout:

Finally logout from the system.

4.2 Sender:

Login:

Sender will login with credentials which are sent by the proxy.

Upload Files:

Only sender upload files after uploading he /she performs file encryption, trapdoor on file to secure it

View Files:

View files which are uploaded by him.

Delete Files:

Finally delete files which are not required to him.

Logout:

Logout from the system.

4.3 Receiver:

Login:

Login with the credentials sent by the proxy.

ISSN: 2233-7857IJFGCN

View Files:
View all files which are uploaded by the sender in servers.
Request:
He requests files which he/she required from the servers after viewing all files.
MyFiles:
View files which are accepted by the proxy.
Logout:
Logout from the system
4.4 Cloud Service provider
Login:
Login with default values.
View Files:
Cloud service provider have access to View all Files from all servers.
Verify Files:
Verifies files which are attacked and secures files in servers .
Logout:
Logout from the site

5. Results and Discussion

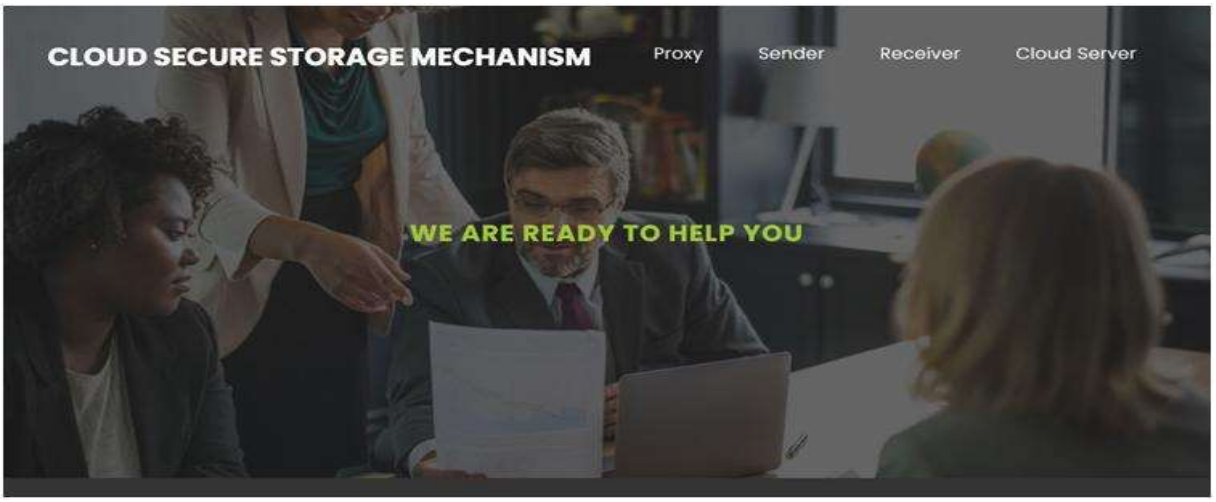

Fig 1: Home Page

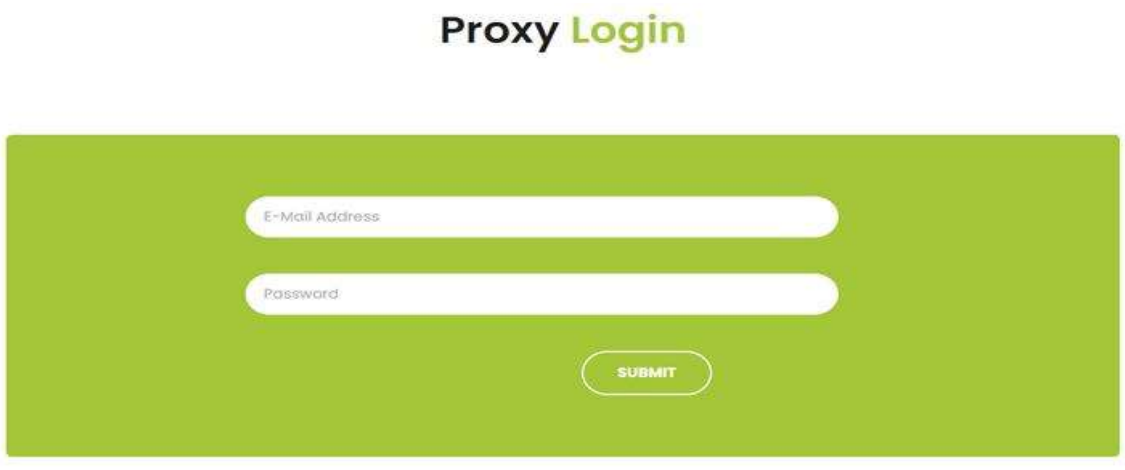

Fig 2: Proxy Login

ISSN: 2233-7857IJFGCN

Fig 3: Client Registration

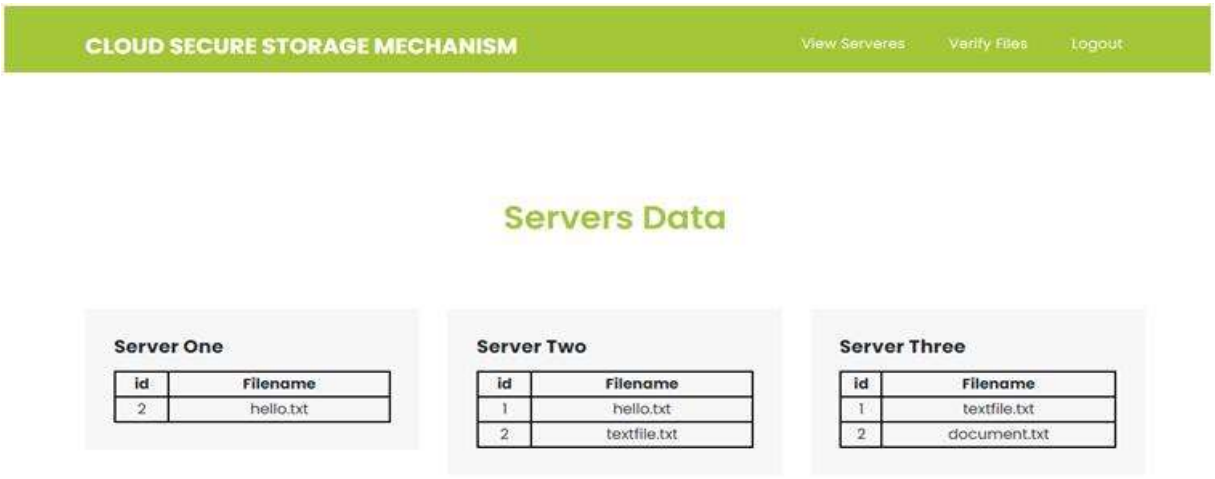

Fig 4: Cloud Secure Storage Mechanism

6. Conclusion and Future Scope

For the issue of cloud data leakage caused by management negligence and malicious attack at storage layer, we proposed CSSM, a cloud secure storage mechanism. CSSM adopted a combined approach of data dispersal and encryption technologies, which can improve the data security and prevent attackers from stealing user data. The experimental results show that CSSM can effectively prevent user data leakage at cloud storage layer. In terms of performance, the increased time overhead of CSSM is acceptable to users. This paper provides a feasible approach to solve the storage security problem, especially prevention from user data leakage at cloud storage layer. CSSM could also effectively protect cryptographic materials from storage perspective.

we will perform the extensive experiments for testing the cloud-based storage with higher volume of data and larger size of access policies in the real cloud environment.

References

[1] A. Bhardwaj, F. Al-Turjman, M. Kumar, T. Stephan, and L. Mostarda, ‘‘Capturing-the-

ISSN: 2233-7857IJFGCN

invisible (CTI): Behavior-based attacks recognition in IoT-oriented industrial control systems,'' IEEE Access, vol. 8, pp. 104956–104966, 2020.

[2] M. Kumar, A. Rani, and S. Srivastava, ''Image forensics based on lighting estimation,'' Int. J. Image Graph., vol. 19, no. 3, Jul. 2019, Art. no. 1950014.

[3] M. Kumar, S. Srivastava, and N. Uddin, ''Image forensic based on lighting estimation,'' Austral. J. Forensic Sci., vol. 51, no. 3, pp. 243–250, Aug. 2017.

[4] J. Li, Y. Zhang, X. Chen, and Y. Xiang, ''Secure attribute-based data sharing for resource-limited users in cloud computing,'' Comput. Secur., vol. 72, pp. 1–12, Jan. 2018.

[5] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, ''Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing,'' Inf. Sci., vol. 379, pp. 42–61, Feb. 2017.

[6] The OpenStack Project. OSSA-2015-006: Unauthorized Delete of Versioned Swift Object. Accessed: Apr. 14, 2015. [Online]. Available: https://security.openstack.org/ossa/OSSA-2015-006.html

[7] The OpenStack Project. OSSA-2015-016: Information Leak Via Swift Tempurls. Accessed: Aug. 26, 2015. [Online]. Available: https://security. openstack.org/ossa/OSSA-2015-016.html

[8] The OpenStack Project. Possible Glance Image Exposure Via Swift. Accessed: Feb. 23, 2015. [Online]. Available: https://wiki. openstack.org/wiki/OSSN/OSSN-0025

[9] Cloud Security Alliance. Top Threats to Cloud Computing: Deep Dive. Accessed: Aug. 8, 2018. [Online]. Available: https://downloads.

cloudsecurityalliance.org/assets/research/top-threats/top-threats-to-cloudcomputing-deep-dive.pdf

[10] The OpenStack Project. OpenStack Security Advisories. Accessed: Feb. 2, 2015. [Online]. Available: https://security.openstack.org/ossalist. html

[11] Common Vulnerabilities and Exposures. CVE-2015-5223. Accessed: Jul. 1, 2015. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename. cgi?name=CVE-2015-5223

ISSN: 2233-7857IJFGCN