JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Cloud Secure Storage Using Block Chain System

DR. A. JYOTHI - Assistant Professor, Department of Computer Science, Anurag University
M. SATHVIKA ANGEL - Student, Department of Computer Science, Anurag University
P. MANIKANTA REDDY - Student, Department of Computer Science, Anurag University
J. SUMANTH - Student, Department of Computer Science, Anurag University

Abstract

The rapid development of cloud computing has been greatly aided by the popularity and power of cloud storage. Nonetheless, there are still serious security incidents that, as a consequence of malicious attack and management negligence, cause widespread sensitive data exposures at the cloud storage tier. To preserve the privacy of cloud data, this study created (CSSM). By combining data dispersion with dispersed storage, CSSM was able to provide encrypted, chucked, and scattered storage, preventing data intrusions at the storage layer. In addition, CSSM used a system of management levels and combined user passwords with secret sharing to avoid any unauthorized access to cryptographic materials. The experimental results show the proposed method can successfully store enormous amounts of cloud data without introducing a substantial latency cost, and is also suitable for securing data at the storage layer from leaking

Cloud storage today depends entirely on large storage providers. Such storage providers function as untrusted third parties that process data for storing, sending and receiving data from an entity. This style of system has many problems, such as high operating costs, software quality and data security. In this paper, we present a model of a multi-user access control system for databases that use blockchain technology to provide stable, distributed data processing. The system allows the data owner to upload the data via a web portal. So, the user who has the secret key to the particular data that has been uploaded to Cloud in encrypted form can only access the folder. Eventually, the system promotes data privacy by maintaining the immutability of the blockchain by processing it in the cloud. We have proposed a secure, blockchain-based data storage and access control system to increase the security of cloud storage.

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 [1], image forensic [2] and forgery detection [3]. As a result, larger volumes of data will be apart of the cloud area. In the cloud industry, Amazon Webservice (AWS) has become the de facto standard. As the core component of the OpenStack that follows this standard, Swift has become one of the most popular cloud storage mechanisms [4], [5]. However, OpenStack Swift mechanism still faces many real security threats [6]– [8] while providing convenient ser-vices. According to Cloud Security Alliance's top threat case analysis report [9] 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 [10] released by OpenStack Vulnerability Management Team VMT, the Swift mechanism may leak user data or configuration information in virtue of security vulnerabilities [11], [12].

1.2 Scope of the Project:

The project scope encompasses the design, implementation, and deployment of a secure cloud storage system, targeting the mitigation of key security threats prevalent in OpenStack Swift mechanisms. This entails integrating RSA encryption for robust key distribution and blockchain technology for data integrity assurance. The project involves developing a system architecture that facilitates seamless interaction between these security measures within cloud computing environments. Implementation efforts will focus on encryption algorithms, blockchain structures, and user interfaces, with rigorous security testing to evaluate system effectiveness and identify vulnerabilities. Optimization strategies will be pursued to manage computational overhead and ensure efficient system performance. Comprehensive documentation and training materials will be provided to support user adoption, with ongoing maintenance and support to sustain system integrity and functionality.

2. LITERATURE SURVEY

Shah et al. [13] 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 [14]– [17] have been adopted to avoid information leakage during machine learning process. Nevertheless, above researches require secure key management mechanisms to prevent cryptographic mate-rials exposure [18], [19]. Zerfos et al. [20] 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 bestowed in memory only. Some approaches [21], [22] intro-ducked 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 [23]. 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. To improve the efficiency and decrease the redundancy, Zheng et al. [25] provided a cloud data deduplication 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

3. OVERVIEW OF THE SYSTEM

3.1 Existing System

AES (Advanced Encryption Standard) has become the encryption algorithm of choice for governments, financial institutions, and security-conscious enterprises around the world. The U.S. National Security Agency (NSC) uses it to protect the country’s “top secret” information. The AES algorithm successively applies a series of mathematical transformations to each 128-bit block of data. Because the computational requirements of this approach are low, AES can be used with consumer computing devices such as laptops and smartphones, as well as for quickly encrypting large amounts of data. For example, the IBM z14 mainframe series uses AES to enable pervasive encryption in which all the data in the entire system, whether at rest or in transit, is encrypted. AES is a symmetric algorithm which uses the same 128-, 192-, or 256-bits key for both encryption and decryption (the security of an AES system increases exponentially with key length). With even a 128-bit key, the task of cracking AES by checking each of the 2128 possible key values (a “brute force” attack) is so computationally intensive that even the fastest supercomputer would require, on average, more than 100 trillion years to do it. In fact, AES has never been cracked, and based on current technological trends, is expected to remain secure for years to come.

3.1.1 Disadvantages of Existing System

Key Management Complexity: The symmetric nature of AES requires the distribution of the same key for both encryption and decryption, leading to significant key management challenges, especially when dealing with multiple recipients or entities.

Key Distribution Vulnerabilities: Distributing AES keys to numerous recipients poses a considerable security risk, as any compromise of the key during distribution could lead to unauthorized access to sensitive data.

Limited Scalability: AES encryption, while efficient for consumer devices and large-scale systems like IBM z14 mainframes, may face scalability issues when deployed in environments requiring frequent key distribution and management.

Susceptibility to Brute Force Attacks: Despite its robust security, AES is theoretically vulnerable to brute force attacks in the long run, as the increasing computational power of supercomputers could potentially reduce the time required to crack AES keys.

Symmetric Encryption Limitations: Symmetric encryption's reliance on a single shared key for both encryption and decryption limit its suitability for scenarios where secure key distribution is challenging, such as in distributed or decentralized systems.

3.2 Proposed System

In proposed system data is secured to store in cloud using two-layer security block chain and encryption for encryption RSA algorithm is used and for Block chain hash bocks are generated with hash key. It is an asymmetric algorithm that uses a publicly known key for encryption, but requires a different key, known only to the intended recipient, for decryption. In this system, appropriately called public key cryptography (PKC), the public key is the product of multiplying two huge prime numbers together. Only that product, 1024, 2048, or 4096 bits in length, is made public. But RSA decryption requires knowledge of the two prime factors of that product. Because there is no known method of calculating the prime factors of such large numbers, only the creator of the public key can also generate the private key required for decryption. RSA is more computationally intensive than AES, and much slower. It’s normally used to encrypt only small amounts of data.

3.2.1 Advantages of Proposed System

Enhanced Key Management: The use of RSA encryption eliminates the key management issue inherent in symmetric algorithms like AES, as it utilizes separate keys for encryption and decryption, improving overall security.

Secure Key Distribution: RSA encryption enables secure key distribution without the need for both parties to share the same key, mitigating the risk of key compromise during distribution.

Blockchain Integration: The integration of blockchain technology provides an additional layer of security, ensuring data integrity and preventing unauthorized access or tampering.

Two-Layer Authentication: The proposed system implements a two-layer authentication method combining encryption and blockchain, enhancing data security and resilience against potential security threats.

Versatility: Despite RSA encryption being more computationally intensive and slower compared to AES, the proposed system is suitable for various applications, including email and file storage across multiple platforms.

3.3 Proposed System Design

In this project work, there are three modules and each module has specific functions, they are:

1. OWNER MODULE

2. USER MODULE

3. CLOUD MODULE

3.3.1 Owner Module:

owner will register into the application by providing all the necessary details and therefore he can login into the application using username and password and user can upload the files to application and share with the other registered users. He can also view the files uploaded by him and can also view the requests for secret key and block chain key from the other users and we can respond and the key and block chain will be sent to user by mail. Using that key, he can download the file and view the information.

3.3.2 User Module:

user will register with application and get user name and password. Owner can see all encrypted files uploaded by all users and send request to respective user and get approval to download data and block chain hash and security keys for RSA are shared to owner email which can be used for owner download.

3.3.3 Cloud Module

Using this module cloud can register with application and store information of each user and owner who uploads and requests for data. Details which are stored in cloud are upload to Drive hq cloud.

3.4 Architecture

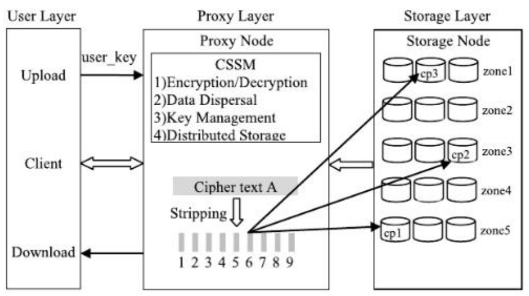
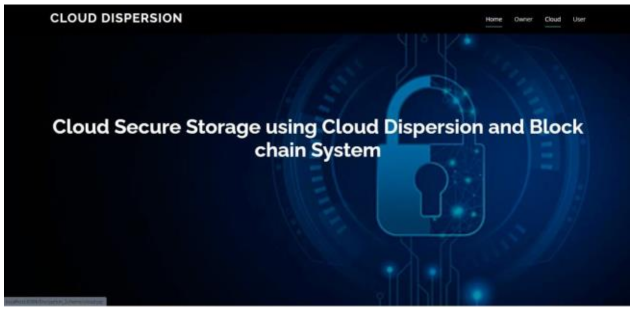
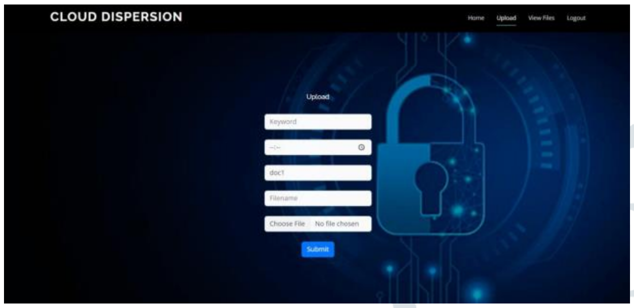
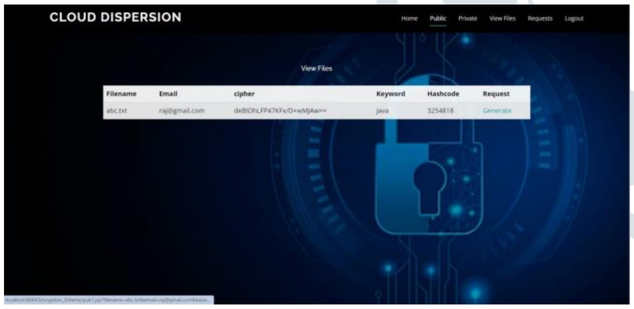


Fig 1: System Architecture

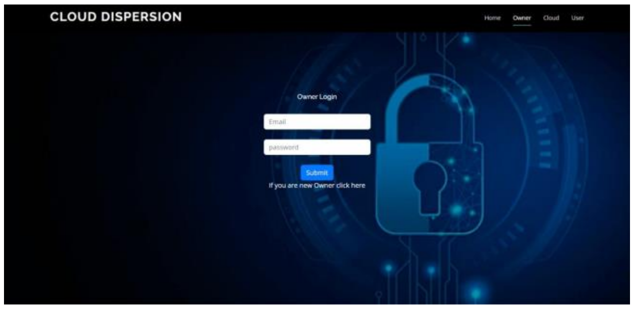
4. RESULT SCREEN SHOTS

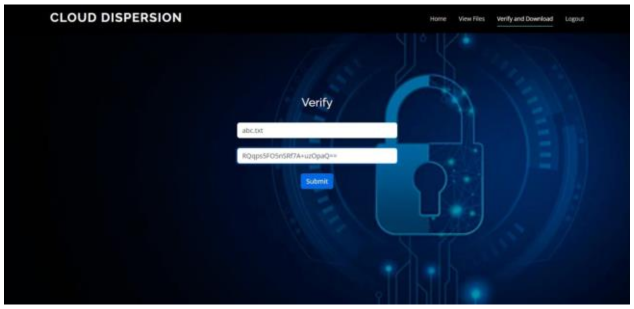

CLOUD DISPERSION
Home
Owner
Cloud
User
Cloud Secure Storage using Cloud Dispersion and Block chain System





Filename	Email	cipher	Keyword	Hashcode	Request
abc.txt	raj@gmail.com	deBIOhLFP47KFx/D+wMJAw==	java	3254818	Generate





abc.txt

RQqps5FO5n5Rf7A+uzOpaQ==

5. CONCLUSION

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 pre-vent 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 stor-age security problem, especially prevention from user data leakage at cloud storage layer. CSSM could also effectively protect cryptographic materials from storage perspective.

6. REFERENCES

[1] A. Bhardwaj, F. Al-Turjman, M. Kumar, T. Stephan, and L. Mostarda, ‘‘Capturing-the-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
[12] Common Vulnerabilities and Exposures. CVE-2016-9590. Accessed: Nov. 23, 2016. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-9590
[13] S. Y. Shah, B. Paulovicks, and P. Zerfos, ‘‘Data-at-rest security for spark,’’ in Proc. IEEE Int. Conf. Big Data (Big Data), Washington DC, USA, Dec. 2016, pp. 1464–1473.
[14] Z. Liu, Y. Huang, J. Li, X. Cheng, and C. Shen, ‘‘DivORAM: Towards a practical oblivious RAM with variable block size,’’ Inf. Sci., vol. 447, pp. 1–11, Jun. 2018.
[15] X. Zhang, X. Chen, J. Wang, Z. Zhan, and J. Li, ‘‘Verifiable privacy preserving single-layer perceptron training scheme in cloud computing,’’ Soft Comput., vol. 22, no. 23, pp. 7719–7732, Dec. 2018.
[16] C.-Z. Gao, Q. Cheng, P. He, W. Susilo, and J. Li, ‘‘Privacy-preserving naive bayes classifiers secure against the substitution-then-comparison attack,’’ Inf. Sci., vol. 444, pp. 72–88, May 2018.
[17] P. Li, T. Li, H. Ye, J. Li, X. Chen, and Y. Xiang, ‘‘Privacy-preserving machine learning with multiple data providers,’’ Future Gener. Comput. Syst., vol. 87, pp. 341–350, Oct. 2018.
[18] B. AlBelooshi, E. Damiani, K. Salah, and T. Martin, ‘‘Securing cryptographic keys in the cloud: A survey,’’ IEEE Cloud Comput., vol. 3, no. 4, pp. 42–56, Jul. 2016.
[19] B. AlBelooshi, K. Salah, T. Martin, and E. Damiani, ‘‘Securing cryptographic keys in the IaaS cloud model,’’ in Proc. IEEE/ACM 8th Int. Conf. Utility Cloud Comput. (UCC), Limassol, Cyprus, Dec. 2015, pp. 397–401.
[20] P. Zerfos, H. Yeo, B. D. Paulovicks, and V. Sheinin, ‘‘SDFS: Secure distributed file system for data-at-rest security for Hadoop-as-a-service,’’ in Proc. IEEE Int. Conf. Big Data, Santa Clara, CA, USA, Oct. 2015, pp. 1262–1271.
[21] J. Zhou, H. Duan, K. Liang, Q. Yan, F. Chen, F. R. Yu, J. Wu, and J. Chen, ‘‘Securing outsourced data in the multi-authority cloud with fine-grained access control and efficient attribute revocation,’’ Comput. J., vol. 60, no. 8, pp. 1210–1222, Feb. 2017.
[22] J. Shao, R. Lu, and X. Lin, ‘‘Fine-grained data sharing in cloud computing for mobile devices,’’ in Proc. IEEE Conf. Comput. Commun. (INFOCOM), Hong Kong, Apr. 2015, pp. 2677–2685.
[23] S. Han, K. Han, and S. Zhang, ‘‘A data sharing protocol to minimize security and privacy risks of cloud storage in big data era,’’ IEEE Access, vol. 7, pp. 60290–60298, 2019.
[24] T. Wang, Y. Mei, W. Jia, X. Zheng, G. Wang, and M. Xie, ‘‘Edge-based differential privacy computing for sensor–cloud systems,’’ J. Parallel Distrib.Comput., vol. 136, pp. 75–85, Feb. 2020.
[25] X. Zheng, Y. Zhou, Y. Ye, and F. Li, ‘‘A cloud data deduplication scheme based on certificateless proxy re-encryption,’’ J. Syst. Archit., vol. 102, Jan. 2020, Art. no. 101666.