Secure Data Migration in Cloud using Encryption

Arulmozhiselvan.L, Research Scholar, Department of Information Science and Technology, Anna University, Chennai & India, arulmozhiselvan@auist.net

Dr.E.Uma, Assistant Professor(SL.GR), Department of Information Science and Technology, Anna University, Chennai & India, umaramesh@auist.net

Abstract **Cloud Computing has many security issues and challenges which are unique in nature. Many methods and algorithms are introduced to solve and overcome the security issues but still the issues and challenges are emerging. The data in the cloud is stored and handled by third party provider and accessed through internet for further updation. So, the data control and perceptibility is limited. The data in cloud should be properly secured by cloud service provider from threats and unauthorized access. The major concern of sensitive data is confidentiality and integrity. Existing methods faced problems in encryption and managing files. This paper proposed secure data storage to avoid data breach. Dispersion of data is integrated to check encryption. Hierarchical approach is introduced to prevent cryptographic attacks. Password method is used to encrypt the file which is unencrypted during updating of file in the cloud. The experimental results show that proposed method provides security to data from attack and has ability to store large amount of data in minimum time. For example, when users upload and download file with our proposed method, it takes only 576seconds/196seconds which is highly efficient.**

***Keywords* — Data Dispersion, Password Encryption, Cloud Computing.**

I. Introduction

With the emerging trend of Internet of things and 5G network, edge devices are utilized to indicate demand in resources needed for AI based edge computing with security enhancement, minimum- bandwidth and low latency. Virtualization in cloud computing plays major role in data augmentation and integration in IT and communication infrastructure. Due to this emerging data sources the resource should be efficient and flexible to access and modify in diverse environment. Cloud-edge-collaborative storage (CECS) [2] act as interface between edge devices and cloud servers. Cloud provides numerous computing resources and storage whereas edge computing has nearly same resources as cloud, but IoT devices have only limited computing resources. Therefore, the edge devices respond to the IoT device request to collect real-time data and other data to be transferred to cloud which save computational cost. The authorized users migrate IoT data to cloud through edge devices.

In this computing, IoT devices collect the data from user and upload data to neighbor edge server. Then, the edge server processes the data to obtain result and stored in cloud server. Finally, users share obtained IoT data on the cloud storage server with respect to request. With the rapid growing of data, the number of business enterprise and individuals migrate data to cloud server. Thus, huge amount of personal, critical data which includes health care, documents, financial data etc., are stored in cloud through internet. It is not ensured that the data stored in cloud is protected from security threats, access control and identity theft. The user send data to cloud for storage and does not know how it is protected by cloud service providers. Data are vulnerable to variety of attacks where the attackers try to steal the personal data for profit.

To overcome these problems and attacks, cryptography helps with encryption. The sensitive data from user should be encrypted before migrating to cloud. The encryption technique protects the data based on key management. Key management and generation used to produce secret keys which are known only to authorized user to update, modify and access. Cryptographic scheme is difficult to design for data that stored in cloud which includes security, efficient way to access and flexibility. Traditional encryption used for application has limited usage because manual solutions are provided. Users must store the keys generated by the encryption algorithm for further updation in data files.

Multiple encryption and decryption of data leads to confusion and user fail to remember the secret keys and data being breached and person identity is compromised. High bandwidth and low latency should be achieved which improves the overall performance. Transparent encryption is adopted in recent applications and implemented based on operating file system. The client interact with cryptographic file and encrypted data are synchronized and connected to cloud servers. It is implemented in kernel which is local or remote system.

Existing system is based on transparent encryption where data are remotely stored in cloud and does not achieve data

DOI : 10.35291/2454-9150.2022.0077

security in multiple environment system. Many cryptographic approaches are restricted in providing most reliable and secure cloud storage because it restricts the user based on fine grained level. In addition, these encrypted files are based on passwords to identify ultimate users to retrieve the secret public and private keys. Other problems in cloud systems includes bottleneck, latency, bandwidth, and failure in single point.

When user wants to decrypt and give access to authenticated users, the data owner interchanges the encryption keys manually, or encrypts individual data with the public keys using asymmetric ciphers. It is not secure to share the secret key with the user because it allows the shared data and personal information to encrypt with the same shared key. When the key is shared with user the owner does not know what the user will do with the key. When the key is breached the data security is compromised. Asymmetric encryption is not exact technique for implementation because the computation and communication cost overhead. To verify and identify user the owner must maintain the public key and must make list of keys for multiple users.

To ensure the stored cloud data many algorithms and methods are proposed which ultimately protects the data in cloud. Identity-based encryption (IBE), attribute-based encryption (ABE), and proxy re-encryption (PRE) are some of the encryption algorithms used in cloud and fog computing. It reduces the complexity of key management complexity for data owners and users to obtain the secret key generated by key generation. However, these encryption techniques use asymmetric encryption to encrypt and decrypt the communication between users and the cloud server, which leads to significant computational complexity.

The proposed work presents a user-side encrypted file system that utilizes the FUSE technology for Linux platforms. Hybrid cryptographic method is proposed which combines symmetric and public key encryption algorithms in order to improve the security and performance of the personal and shared files that are migrated. Symmetric encryption is used to encrypt the files to reduce the encryption cost which the asymmetric algorithms are used to secure the interchange of secret keys for the files. The goals of the proposed method are twofold. First, design a cryptographic layer that effectively encrypts all files that are outsourced to the cloud storage in a highly secure and transparent manner. Second, enable a secure data sharing of cloud storage at the granularity of individual files based on IBE scheme that is combined with the proposed method.

II. Related Work

Cloud provides effective way to share data from multiple users from different location and time. Users send sensitive data which is highly confidential and does not trust third party service to maintain data. So, the stored data should be secure and efficient way to access. In order to protect data from attackers, identity-based encryption algorithm is proposed. Revocable storage identity-based encryption [1] also implemented to support key generation and management in cloud storage system. The data storage and maintenance are the issues faced by the cloud system, to overcome the issue many integrity auditing protocols are introduced. Data integrity auditing protocol [2] is proposed to secure and store data without the use of private key. The user generate token which is hardware based to store private keys. The key is activated to generate password for the file to be stored in cloud.

Various attacks are performed to steal the confidential and sensitive data stored in cloud. The data should be protected from this malicious attack with the help of cryptographic methods. The data dispersion with password method [3] is proposed to secure data and detect the attack. Encryption integrated with search option provides confidentiality and preserve privacy of data. Regular language search encryption [4] is proposed which protects the data from keyword guessing attack. Regular language search generates key which is private used to communicate with key generator which is efficient and secure. To check integrity of data stored in cloud storage server, new method is proposed used to ensure secure storage of data. SEPDP [5] preserve privacy of data in cloud and maintains integrity.

The integrity auditing protocol [6] is introduced to secure the cloud data. The data migrated to cloud consists of sensitive and confidential data which is to be highly protected from attacks. Various algorithms are proposed to secure and protect cloud data but fails to maintain confidentiality and integrity. Audit protocol consists of algebraic operations which is efficient way to audit the data at regular interval of time. Ternary hash tree-based integrity [7] is proposed for secure cloud data storage using verification of integrity. Cloud service provider in cloud is not trusted by user due to third party nature. So, auditor verify the data integrity which reduces computation cost. Hash-based tree methodology is used to audit the data which is stored in cloud for further use in future.

Cloud-edge storage [8] is proposed to process IoT data which is collected through edge devices. Due to vulnerable attack in cloud servers, data is breached easily by attackers. Existing system consists of various algorithm which secure only the trusted servers not all storage server located in cloud. The proposed method provides both private and public key to secure data from unauthorized access. Auditing protocol for regenerating code [9] is introduced to audit the data for security and privacy from various cryptographic attacks. Its secure data by delegating proxy of user to authenticate cloud storage blocks for security purpose.

Attribute authority management [10] scheme is proposed to access control and reduce the risk of data storage using public keys. The proposed method contains both backward and forward security attribute data where user ensure the data uploaded is secure. The data stored in cloud faces denial of access, confidentiality, and integrity in hybrid environment. So, service-oriented solution [11] is provided to secure data storage. It helps to store and retrieve files securely and efficient. To protect the cloud storage system that has multiple tasks to perform is achieved by proxy encryption. AES and proxy re-encryption [12] are introduced to encode file and migrate to cloud in encrypted format. Erasure encode is utilized in distributed cloud storage system.

DOI : 10.35291/2454-9150.2022.0077

To reduce operating cost and improve utilization of resource, IaaS provide solution [13]. It does not allow the administrator to access or breach data at any cost. Malicious attack performed by administrator is also reduced in IaaS infrastructure. Homomorphic hash algorithm [14] is introduced to reduce the communication time and computation cost of data stored in cloud. Deduplication methodology is also used to check the existence of file and reduce the storage space. Data security in cloud environment comprises multiple storage system based on traditional cloud system. Erasure code [15] is utilized to stop original file which migrate to cloud to encrypt the file with the help of AES. It overcomes the existing problem of data security and improves the reliability of multiple cloud storage system.

Verification of data file stored in cloud is important to process the query that used to obtain the file and access it. Fine-Grained query [16] is used to verify the data file at every query stage which authenticate the user who wants to access the file. It protects the data from SQL hijacking which steals the entire database. Charon [17] cloud-based storage system is proposed to enhance the performance of file system. The computational cost is reduced by conflicts between user who supposed to access the file and perform operations. Channel information is breached when uploading file in cloud. Dispersed Convergent Encryption [18] is proposed to secure the cloud data which has deduplication protocol that does not involve any third party to audit. It overcome the problem of data security and channel side template attack.

Integrity of data is still challenge in cloud storage system which is important in outsourcing data. Attribute-based auditing protocol [19] is proposed to upload data files in cloud based on certain attribute types and can assign auditor of their own to check the outsource data from cloud and user. Efficient auditing scheme [20] is needed to secure the data in both cloud and fog computing servers. Based on MAC and HMAC, the auditing is designed using private key. Usage of public key cryptographic encryption causes communication and computation cost overhead. To reduce and improve, private key is used. Constant problem in cloud storage is ensuring authentication [21] and access the data from secure storage environment. Proxy re-encryption [22] is utilized which is secure and efficient way to protect data from impersonation attacks. The data access is controlled by user and generates key to access that data by another user or client. Cipher text attribute-based encryption [23] is proposed to authorize and authenticate user from various attacks.

III. PROPOSED FRAMEWORK

The emerging technology includes Internet of Things (IoT), smart world, digital India transformation at the top. The data from different fields are migrated to cloud for secure storage. Like population increases, data also increased in massive amount which needs space for storing the entire data. Unauthorized access, data breach, identity leakage and privacy data breach increase even it is encrypted and stored in cloud. The data security should be provided by cloud service provider. When the data is migrated to cloud, security issues arise. The data encryption is to convert original text into 0's and 1's which is unassumed by cryptographic algorithms. If someone try to access or steal the original data from file, it is impossible which ultimately protects the data confidentiality and secure data from attackers. Only the authorized user decrypt files to access the original file. Encryption can be symmetric and asymmetric which uses key to encrypt and decrypt files. Encryption of file consists of many types which is chosen based on the data type and security level. Identity based, attribute based, homomorphic encryption is some of the types which secure data in the file by encrypting with secret key generation.

To improve confidentiality of client data in cloud, secure storage mechanism is proposed which protects and secure data from security breaches. The main objective is to make difficulty for attackers who tries to steal data. Data dispersion is used to segment the uploaded files into several parts. When the client/user uploads the files in cloud, it is segmented and stored in different nodes. This makes the attackers to steal only minimal data not the entire file from the cloud storage server. Though the files are segmented it is possible to retrieve complete data by attacker with the help of logical relationship between the data. So, encryption is proposed which further secure data from attackers.

To protect data from attackers, data dispersion is used which segment the file and stored in different cloud storage nodes. Uncertainty of location is difficult for attackers to locate entire segmented files and retrieve original file. When the user wants to access the file, it needs to recover the segmented file based on logical relationship between user and segmented file. In between the process, it is possible to steal the data which is stored in proxy nodes. The above-mentioned security attacks can be overcome by trusted environment. The proxy node should adopt integrity and ensure whether the code is secured and trusted. Next, the environment should check whether the code runs without any interference.

Encryption is the most traditional and common methods to secure user data from attackers. To reduce time, symmetric encryption is adopted which uses key for encrypting and decrypting files.

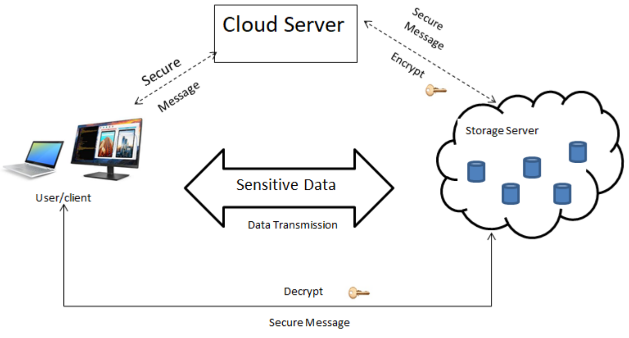

Fig 1: Architecture of secure storage in cloud

Key management is important to consider in order to protect keys which is used for encryption and decryption process. In cloud, usually the segmented files are stored at different nodes or storage container where it is assigned

DOI : 10.35291/2454-9150.2022.0077

with symmetric key. The container or node is integrated to form box which has separate key to access it. Here, the key is set by the user so that only the user knows the exact key to access. If the attacker wants to steal two keys are required which is again difficult task to steal data from cloud.

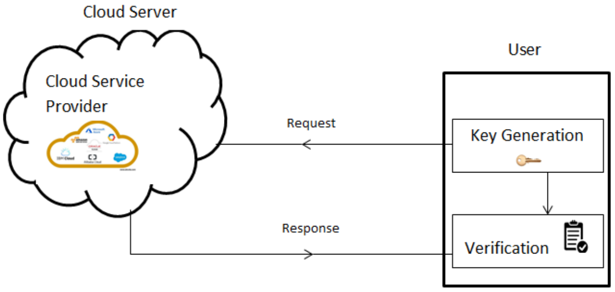

Fig 2: Data integrity in storing data in cloud

When the user wants to download or access the file, request is initiated to get password through the API interface. The key generator on proxy node ensures user has both the keys to access and download the file from the cloud storage container. The container key is required to recover the segmented files which are generated by hierarchy derivation algorithm. The required files are recollected from the cloud container to decrypt based on user key and finally the entire file is returned to user for further process.

IV. Experimental Result

The environmental setup consists of proxy nodes and other storage nodes in cloud. Each storage nodes consists of two disk which is used for data and system. The proposed method improves security at proxy nodes, encryption decryption process and key management. The encrypted data file is stored in one container with password protection. Each container consists of ten blocks which contains cipher text. The container nodes are further processed by secret sharing algorithm to secure and protect data from attackers with help of keys. The algorithm generates threshold values where the data is encrypted and stored in blocks.

The finite field is used to distribute secret keys which is randomly taken from finite field of non-zero elements. The key recovery process is generated to obtain the block keys in order to retrieve the original files. When the attacker accesses the encoded block, the key generated has threshold values which preserve the security of data.

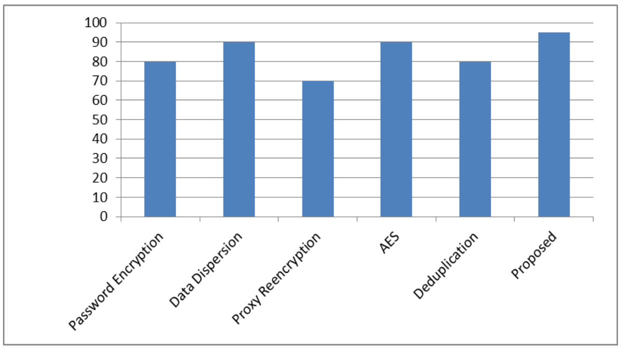

Fig 3: Comparison between various algorithm

The performance analysis is based on time and space complexities. The proposed method uses AES encryption which is directly proportional to size of file being encrypted. On comparing the size of user file, the time needed for encryption and decryption is comparatively small. Hierarchy derivation algorithm is adopted where the user can set the key for their blocks. By this comparison, it is proved that the time cost is small, and it can be ignored. The space required for storing the file is proportional to number of files the user needs to upload. The storge space of container node is proportional to total number of storage containers in cloud. Compare to size of files which the user needs to upload, the storage is small. To evaluate the performance of proposed method, time is compared with upload and download of files in cloud system. If the file size is between 42KB to GB. Time considered to complete each file, it is possible to complete ten task which upload and download file in cloud. The time consumption is average.

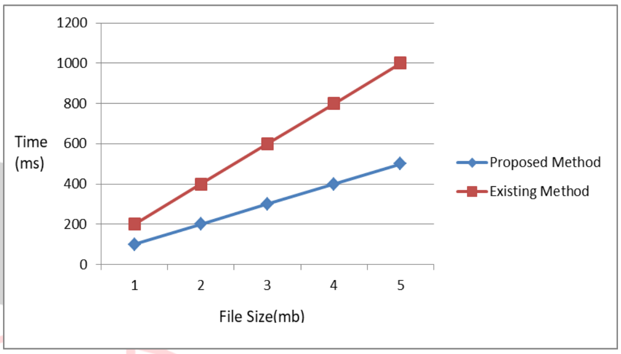

Fig 4: Computation

As the file size increases, the time required to upload and download file also increases. While uploading file in cloud system it uses concurrent method in encryption and data dispersion. When downloading file, it is required to assign the location for each segmented file. Once the segmented files are collected, the original file is recovered in sequential order.

V. Conclusion

Data effluence from cloud by management and malicious attack causes issues in storing data securely. The proposed method secures the data and protects data from attack. Data dispersion and encryption integrated to provide security for data and secure from attackers who steal data from cloud. The file to be stored in cloud is encrypted to secure file based on file per keys. It uses both symmetric and asymmetric encryption methods for file encryption before uploaded in cloud. It only allows the authorized user to access and upload file with secret keys, achieves integrity. Tampering and deletion attacks are overcome by encrypting the file from user side. The experimental results show that proposed method prevents the data effluence at cloud storage. The performance indicates that it uploads and download file efficiently in minimum time which satisfies the user in terms of time and security. In terms of performance, the increased time overhead is acceptable to users. It also overcomes brute force attack, man-in-the-

DOI : 10.35291/2454-9150.2022.0077

middle attack on file stored in cloud. To enhance performance further, parallel encryption can be adopted in future.

REFERENCES

[1] Kwangsu Lee, Wenting Shen and Jia Yu, "Comments on Secure Data Sharing in Cloud Computing Using Revocable-Storage Identity-Based Encryption," IEEE Transactions on Cloud Computing, Vol. 8, no. 4, pp. 1299-1300, 2020.

[2] Jing Qin, Jia Yu and Rong Hao, "Data Integrity Auditing Without Private Key Storage for Secure Cloud Storage," IEEE Transactions on Cloud Computing, Vol. 9, No. 4, pp. 1408-1421, 2021.

[3] Heqing Song, Jifei Li and Haoteng Li, "A Cloud Secure Storage Mechanism based on Data Dispersion and Encryption," IEEE Access, Vol. 9, pp. 63745-63751, 2021.

[4] Yang Yang, Xianghan Zheng, Chunming Rong and Wenzhon, "Efficient Regular Language Search for Secure Cloud Storage," IEEE Transactions on Cloud Computing, Vol. 8, no. 3, pp. 805-818, 2020.

[5] Jia Ju and Rong Hao, "Comments on SEPDP: Secure and Efficient Privacy Preserving Provable Data Possession in Cloud Storage," IEEE Transactions on Services Computing, Vol. 14, no. 6, pp. 2090-2092, 2021.

[6] Yang Yang, Yanjiao Chen and Fei Chen, "A Compressive Integrity Auditing Protocol for Secure Cloud Storage," IEEE/ACM Transactions on Networking, Vol. 29, no. 3, pp. 1197-1209, 2021.

[7] M. Thangavel and P. Varalakshmi, "Enabling Ternary Hash Tree Based Integrity Verification for Secure Cloud Data Storage," IEEE Transactions on Knowledge and Data Engineering", Vol. 32, no.12, pp. 2351-2362, 2020.

[8] Ye Tao, Peng Xu and Hai Jin, "Secure Data Sharing and Search for Cloud-Edge-Collaborative Storage," IEEE Access, Vol. 8, pp. 15963-15972, 2019.

[9] Jindan Zhang, Rongxing Lu and Baocang Wang, "Comments on Privacy-Preserving Public Auditing Protocol for Regenerating Code Based Cloud Storage," IEEE Transactions on Information Forensics and Security, Vol. 16, pp. 1288-1289, 2020.

[10] Shuming Xiong, Qiang Ni, Liangmin Wang and Qian Wang, "SEM-ACSIT: Secure and Efficient Multiauthority Access Control for IoT Cloud Storage," IEEE Internet of Things Journal, Vol. 7, no. 4, pp. 2914-2927, 2020.

[11] Surya Nepal, Carsten Friedrich, Leakha Henry and Shiping Chen, "A Secure Storage Service in the Hybrid Cloud," Fourth IEEE International Conference on Utility and Cloud Computing, pp. 24-29, 2011.

[12] R. Nivedhaa and J. Jean Justus, "A Secure Erasure Cloud Storage System using Advanced Encryption Standard Algorithm and Proxy Re-Encryption," 2018 International Conference on Communication and Signal Processing, pp. 115-124, 2018.

[13] Jinho Seol, Seongwook Jin and Seungryoul Maeng, "Secure Storage Service for IaaS Cloud Users," 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 13-16, 2013.

[14] R. Patil Rashmi, Yatin Gandhi, Vinaya Sarmalkar and Prajakta, "RDPC: Secure Cloud Storage with Deduplication Technique," Fourth International Conference on ISMAC, pp. 7-16, 2020.

[15] Wei Shi, Tenglong Liu and Min Huang, "Design of File Multi-Cloud Secure Storage System Based on Web and Erasure Code," IEEE 11th International Conference on Software Engineering and Service Science (ICSESS), pp. 16-25, 2020.

[16] Hui Yin, Zheng Qin, Jixin Zhang, Lu Ou and Keqin Li, "Achieving Secure, Universal and Fine-Grained Query Results Verification for Secure Search Scheme Over Encrypted Cloud Data," IEEE Transactions on Cloud Computing, Vol. 9, no. 1, pp. 27-39, 2021.

[17] Ricardo Mendes, Tiago Oliveira, Vinicius Cogo and Nuno, "Charon: A Secure Cloud-of-Clouds System for Storing and Sharing Big Data," IEEE Transactions on Cloud Computing, Vol. 9, no. 4, pp. 1349-1361, 2021.

[18] Yuan Zhang, Yunlong Mao, Minze Xu and Fengyuan Xu, "Towards Thwarting Template Side-Channel Attacks in Secure Cloud Deduplications," IEEE Transactions on Dependable and Secure Computing, Vol. 18, no. 3, pp. 1008-1018, 2021.

[19] Yong Yu, Yannan Li, Willy Susilo and Guomin, "Attribute-Based Cloud Data Integrity Auditing for Secure Outsourced Storage," IEEE Transactions on Emerging Topics in Computing, Vol. 8, no. 2, pp. 377-390, 2021.

[20] Xingjun Zhang and Wei Si, "Efficient Auditing Scheme for Secure Data Storage in Fog-to-Cloud Computing," IEEE Access, Vol. 9, pp. 37951-37960, 2021.

[21] Zahid Ghaffar, Shafiq Ahmed and Khalid Mahood, "An Improved Authentication Scheme for Remote Data Access and Sharing Over Cloud Storage in Cyber-Physical-Social-Systems," IEEE Access, Vol. 8, pp. 47144-47160, 2020.

[22] Han Qiu, Hassan Noura, Meikang Qiu and Zhong Ming, "A User-Centric Data Protection Method for Cloud Storage Based on Invertible DWT," IEEE Transactions on Cloud Computing, Vol. 9, no. 4, pp. 1293-1304, 2021.

[23] Jianting Ning, Zhenfu Cao, Xiaolei Dong and Kaitai Liang, "CryptCLoud+: Secure and Expressive Data Access Control for Cloud Storage," IEEE Transactions on Services Computing, Vol. 14, no. 1, pp. 111-124, 2018.

DOI : 10.35291/2454-9150.2022.0077