

Secured Efficient Data Access in Cloud Architecture with B+ Tree Algorithm

Nagajothi.A¹, M.Akilandeswari², S.Meenakshi³

Assistant Professor, Department of CSE, Karpagam College of Engineering, Coimbatore, Tamilnadu, India. ¹²³

ABSTRACT: Cloud computing facilitates data protection, privacy and medical record access. Computing is the new trend to reduce the investment in business to satisfy the client needs using internet. The various cloud services provides infrastructure, software and platform. In cloud computing, data is moved to a remote location. Users store their data conscientiously in the cloud and return back when it is needed. But there is no assurance for the data stored in the cloud is secure and not changed by the cloud or Third Party Auditor (TPA). Cloud storage architecture will have a collection of storage servers with higher end configuration which will provides long-term storage services over the Internet and also for the cloud storage system. Here storing and retrieving the data in a third party's cloud system causes serious problems and conflict over data confidentiality during the data transactions. Whenever third party storage will involved with the cloud server this conflict will occur naturally. But general encryption processor and verifier schemes protect data confidentiality during the transaction of dual execution. In this paper, we propose a secured threshold proxy re-encryption server and integrate it with a decentralized erasure code such that a secure distributed storage system is formulated. The main technical contribution is that the proxy re-encryption scheme supports encoding operations along with a key over encrypted messages and forwarding operations over encoded and encrypted messages. This method is implemented for secured data forwarding. During data forwarding a proxy server will be created virtually to access the encrypted data from the sender side. This makes less traffic and the original data content will not get affected during the time of data transaction. After the transaction the proxy server will be deleted. In this paper, an enhanced method for accessing the user data by B+ tree is proposed. With the help of B++ tree algorithm, the TPA will access the data at the CSP of the time complexity of $O(\log n)$. In turn, the user can access their data in an efficient manner.

KEYWORDS: B++ tree, encryption, data forwarding, decryption and Proxy, TPA and CSP.

I. INTRODUCTION

Basically cloud storage architecture will have a collection of storage servers with higher end configuration which will provides long-term storage services over the Internet and also for the cloud storage system. Proxy re-encryption scheme [5] provides security improvements over other approaches used earlier. The main advantage of this scheme is that they are unidirectional and do not require delegators to reveal their entire secret key to anyone or even interact with the delegate, in order to allow a proxy to re-encrypt their cipher texts. In schemes, only a limited amount of trust is placed in the proxy. For example, it is not able to decrypt the cipher texts it re-encrypts, and we prove our schemes secure even when the proxy publishes all the re-encryption information it knows. This enables a number of applications that would not be practical if the proxy needed to be fully trusted. At the early years, the Network-Attached Storage (NAS) and the Network File System (NFS) provide extra storage devices over the network such that a user can access the storage devices via network connection. Afterward, many improvements on scalability, robustness, efficiency, and security were proposed. Here storing and retrieving the data in a third party's cloud system causes serious problems and conflict over data confidentiality during the data transactions. Whenever third party storage will involved with the multi cloud server this conflict will occur naturally. Even though there are various methods are available to overcome this problem like cryptography, key encryption and etc. But general encryption processor and verifier schemes protect data confidentiality during the transaction of dual execution, but along with this process the main drawback will, it limits the functionality of the storage system. This is because a few operations only supported over encrypted data. These methods will cause failure. In order to constructing a secure storage system that supports multiple functions is challenging when the storage system is distributed and has no central authority. The application logic proposes a secured threshold proxy re-encryption server and integrates it with a decentralized erasure code such that a secure distributed storage system is formulated. In this method multiple users can interact with the storage system. Users can upload their data in to the distributed storage system. The distributed storage system not only supports secure and robust data storage and retrieval, but also lets a user forward his data in the storage servers to another user without retrieving the data back. This makes the ownership data unused and secured during the time of retrieval. The main technical contribution is that the proxy re-encryption scheme supports encoding

operations along with a key over encrypted messages, as well as forwarding operations over encoded and encrypted messages. The content in the database will be in the decrypted format. So that even intruder cant able to access the data even they access the database. The encrypted data will become unused even the data obtained by the intruder. This makes the system so stronger. This project deals with fully integrates encrypting, encoding, and forwarding. The application can be shown in both cloud servers as well as in local host as per the environment. The storage and robustness are more flexible with the users. So that user will authorize the sender request to generate the key. Using the authorized one time key sender can access the encrypted file in decrypted format at once. The key will become invalid after one use. This is method is implemented for secured data forwarding. During data forwarding a proxy server will be created virtually to access the encrypted data from the sender side. The original data from the cloud server will be transmitted to the proxy virtually. This makes less traffic and the original data content will not get affected during the time of data transaction. After the transaction the proxy server will be deleted.

II. LITERATURE SURVEY

Lin et al. [12] are defined “A Secure Decentralized Erasure Code for Distributed Network Storage,” Decentralized Erasure Codes are linear codes with a specific randomized structure inspired by network coding on random bipartite graphs, they are optimally sparse, and lead to reduced communication, storage and computation cost over random linear coding. Kallahalla et al. [11] defined “Plautus: Scalable Secure File Sharing on Untrusted Storage”, use of cryptographic primitives to protect and share files. Plautus features highly scalable key management while allowing individual users to retain direct control over who gets access to their files and mechanisms in Plautus to reduce the number of cryptographic keys exchanged between users by using file groups, distinguish file read and write access, handle user revocation efficiently, and allow an untrusted server to authorize file writes. Lin et al. [10] defined “A Secure Erasure Code-Based Cloud Storage System With Secure Data Forwarding” A decentralized erasure code is suitable for use in a distributed storage system , after the message symbols are send to storage servers each storage server independently computes a codeword for received message symbols and stores it. Ateniese et al. [2] proposed “Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage” that present new re-encryption schemes that realize a stronger notion of security, and demonstrate the usefulness of proxy re-encryption as a method of adding access control to a secure file system. Performance measurements of our experimental file system demonstrate that proxy re-encryption can work effectively in practice.

III. SYSTEM MODEL

a. Encryption and Decryption:

This is used to encrypt the plain text into a cipher text. Cipher text is produced along with a single key. This is used to convert the cipher text again into plain text. The data is encrypted with single key using random key generation algorithm. Storing data in a third party does not provide confidentiality in cloud storage. Data confidentiality is provided by proxy re-encryption scheme. In order to overcome the security problem that is incurred by the intruders, the key is encrypted at the user site and is shared with the TPA to have a secured data communication.

b. Proxy re-encryption in privacy scheme

Encryption technique is used to convert plain text into cipher text. Proxy re-encryption provides data confidentiality in cloud storage system. Decentralized erasure code is used to compute codeword for each message symbol. Proxy re-encryption plays important role in the privacy scheme. This is because request from another user will accept by cloud server, as well as the cloud server will produce another request to data owner. After the authorizing the request from the data owner a 64 bit key will be generated for data visualization. Here proxy re encryption will be successfully implemented in the privacy scheme. The encrypted key will in the pending request, until another user accesses the key to view the data of the data owner. Proxy re encryption keys is one use key. So that keys cannot be duplicated. Moreover in case of hacking the key it will take nearly 18 days to 45 days. But the new user will use the key within time. Else the key request will be deleted.

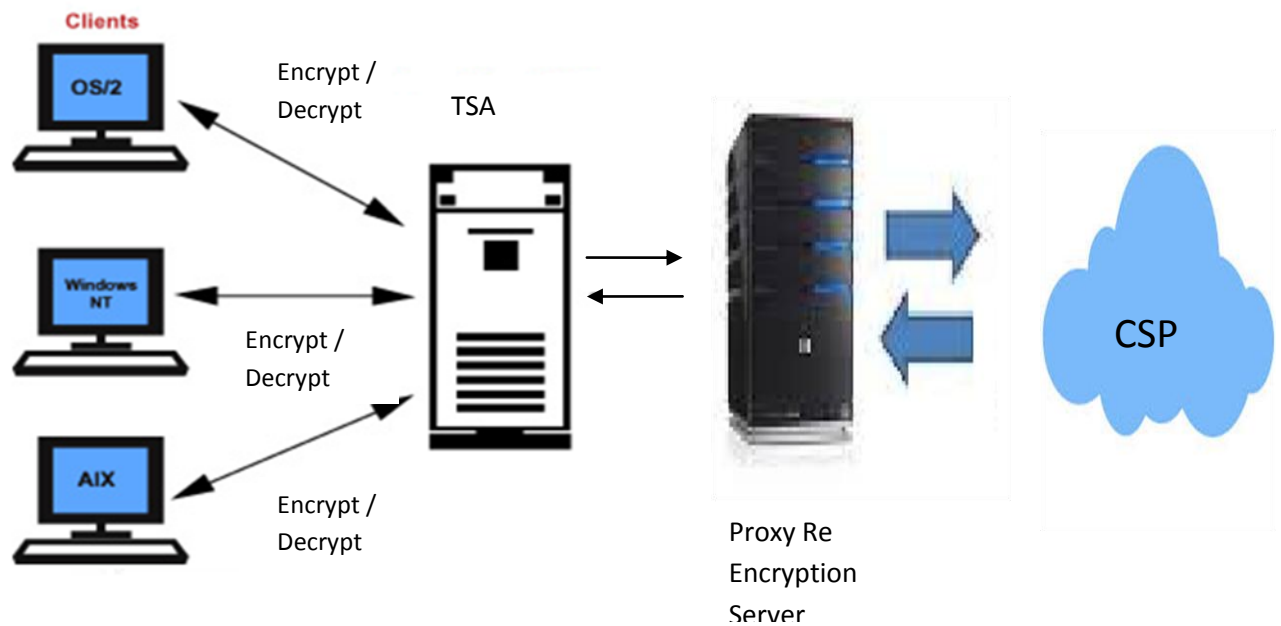


Fig1. Illustration of Proposed System Architecture.

Data owner Store the original data into the cloud storage server. The data will be encrypt and stored in the cloud storage server. Anyone can view the uploaded data. But the data will be in the encrypted format. Another user can only view the file name of the data. And another user will send request to the cloud server through the TPA to view the data. TPA will forward the request from the user to the CSP. Data owner wants to accept the request from the cloud server. Cloud server will forward a de encrypted key to the user. Simultaneously cloud server will create a virtual server and decrypts the data from the cloud storage server. User can enter the de encrypted key to view the original data. The key will be valid for one time only. After the verification of the decrypted data then the User can view the data at their site.

In the multiuser environment, there is a possibility for the users to wait for a long time in order to access their own data. This overhead occurs due to the encryption and decryption process which have to be done simultaneously to more number of users requesting their service from the same CSP. To overcome this problem, the proposed system makes use of B++ tree algorithm to search the users' data in their databases as well as file systems. Whenever the request comes to the TPA from the user regarding their data access, the TPA will get the user ID and the Encrypted key. Once the TPA authenticate the user with the valid input, the user id is given as input to B++ tree Processor in order to access the data at the time complexity of $O(\log n)$.

The proposed system provides three levels of security to the user data. The first level of security is to have a TPA between the CSP and the User. Since the TPA itself should not bring any new vulnerabilities, the second level of security is provided which is the introduction of encryption and decryption processor between the TPA and the User to perform crypt analysis on data. The proposed system uses the proxy re encryption method, which is a third stage of providing security to the user data.

IV. CONCLUSION

The proposed system makes use of various levels of security to the user data with proxy encryption server, third party auditor and the encryption processor.

The searching operation is made easier with the help of B++ tree at the CSP.

The data access time will be reduced less than $O(\log n)$ if order of B++ tree is increased further. The order of B++ tree is fixed based on the services provided by the CSP and the Number of users requesting the service from the CSP.

REFERENCES

- [1] J. Shao and Z. Cao, "CCA-Secure Proxy Re-Encryption without Pairings," Proc. 12th Int'l Conf. Practice and Theory in Public Key Cryptography (PKC), pp. 357-376, 2009.
- [2] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Trans. Information and System Security, vol. 9, no. 1, pp. 1-30, 2006.
- [3] 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), pp. 598-609, 2007.
- [4] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. Fourth Int'l Conf. Security and Privacy in Comm. Netowrks (SecureComm), pp. 1-10, 2008
- [5] G. Ateniese, K. Benson, and S. Hohenberger, "Key-Private Proxy Re-Encryption," Proc. Topics in Cryptology (CT-RSA), pp. 279-294, 2009.
- [6] R. Bhagwan, K. Tati, Y.-C. Cheng, S. Savage, and G.M. Voelker, "Total Recall: System Support for Automated Availability Management," Proc. First Symp. Networked Systems Design and Implementation (NSDI), pp. 337-350, 2004.
- [7] M. Blaze, G. Bleumer, and M. Strauss, "Divertible Protocols and Atomic Proxy Cryptography," Proc. Int'l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT), pp. 127-144, 1998.
- [8] A.G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Ubiquitous Access to Distributed Data in Large Scale Sensor Networks through Decentralized Erasure Codes," Proc. Fourth Int'l Symp. Information Processing in Sensor Networks (IPSN), pp. 111-117, 2005.
- [9] A.G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Decentralized Erasure Codes for Distributed Networked Storage," IEEE Trans. Information Theory, vol. 52, no. 6 pp. 2809-2816, June 2006.
- [10] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "Plautus: Scalable Secure File Sharing on Untrusted Storage," Proc. Second USENIX Conf. File and Storage Technologies (FAST), pp. 29-42, 2003.
- [11] H.-Y. Lin and W.-G. Tzeng, "A Secure Decentralized Erasure Code for Distributed Network Storage," IEEE Trans. Parallel and Distributed Systems, vol. 21, no. 11, pp. 1586-1594, Nov. 2010.
- [12] M. Mambo and E. Okamoto, "Proxy Cryptosystems: Delegation of the Power to Decrypt Cipher texts," IEICE Trans. Fundamentals of Electronics, Comm. and Computer Sciences, vol. E80-A, no. 1, pp. 54-63, 1997.
- [13] Q. Tang, "Type-Based Proxy Re-Encryption and Its Construction," Proc. Ninth Int'l Conf. Cryptology in India: Progress in Cryptology (INDOCRYPT), pp. 130-144, 2008.

BIOGRAPHY



A. Nagajothi received her B.E` degree in Computer Science and Engineering from Karpagam College of Engineering, India, in 2005, the M.E. degree in Computer Science and Engineering from Anna University, in 2010. She is working as an Assistant Professor in the Department of Computer Science and Engineering, Karpagam College of Engineering, Coimbatore. Her research interests include Neural Networks, Artificial Intelligence, Cloud computing, Grid Computing and Image Processing.