SECRET KEYS AND THE PACKETS TRANSPORTATION FOR PRIVACY DATA FORWARDING METHOD IN CLOUD SERVER

P. Velavan[1], K. Balaji[2], R.Ganesh[3]

[1]Assistant Professor, Dept of CSE, Apollo Engineering College, Chennai, India
[2]Student, Dept of CSE, Apollo Engineering College, Chennai, India
[3]PG Scholar, A.R.J.I.T, Tamilnadu, India

Abstract

The Cloud computing is the process of storing the data in the Remote server. This process doesn't speak much about confidentiality and robustness of the data. To improve the security and confidentiality the uploaded file from a data owner is splitted into multiple packets and stored in multiple cloud servers. These packets are encrypted using the primary key. These different keys are also distributed in multiple key servers. User id is appended for verification. If the data owner forwards the file then the keys are verified for the data access. In this we are proposing sending the secret key as SMS to the shared or forwarded nodes for the process of proper Security. This technique integrates the concepts of encryption, encoding and forwarding.

Keywords-*cloud computing, encryption, storage system*

----------***----------

1. INTRODUCTION

Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage The idea of cloud computing is based on a very fundamental principal of ―reusability of IT capabilities Cloud computing is one of the hottest trends in the industry. This has proven as a bliss for small business and enterprise-sized IT. Cloud hosting has considered as the major shift in the way companies use to look their IT infrastructure. This type of approach to IT relies on the Internet, and usually involves provisioned, scalable, dynamic and virtual solutions. Cloud computing pulls the details of IT infrastructure management away from the business and puts it squarely in the hands of true experts.

Storing the data in the cloud server makes serious concern on data confidentiality .To avoid that ,the user can encrypt the messages using encryption algorithms before applying the erasure code method to encode and store messages. The erasure code will reduce the expansion times of encoded messages [5],[7].

In this paper we are proposing the security concerns in the data forwarding scheme. The data owner will send the data to the cloud server .The cloud server will receive the data and do tDhe encryption for the data. The data owner holding the private key and pubic key pair. The encrypted data will be stored in data server and key servers separately. The data packets will be stored in the randomly chosen data servers and the generated keys will be stored in the key servers. If any authorized user want to modify them data in the cloud server, they will send the request to cloud server by appending user-id as an verification code.

After verification is done by the server side, the cloud server will give the permission to the data owner to modify the data. If the data owner wants to forward the data to the other data owner, they can forward. But the new one will receive the encrypted data only. They can't view the data without the decryption process. For the decryption process the data owner will send the key for the who got the encrypted data recently. For the secured transformation of the key, that will be sending by an SMS. So, the intruder can't aware of the secret key transformation.

The decentralized architecture gives better scalability and robustness rather than the existing architectures in the Distributed environment [1][3]. These erasure codes are random linear codes with specific randomized structure of network. The proxy re-encryption scheme scores best granularity on the granted right of the key servers [2]. Accomplishing the integration with consideration of distributed environment is more challenging task. Cloud has been used as the metaphor for the internet computing.

The cloud can be categorized into SaaS(service as a software), IaaS (Infrastructure as a service) & PaaS (Platform as a Service). While SaaS is by far the most common type of cloud computing implementation today, other types are rapidly gaining popularity as companies see the cost and expertise advantages of each. It also make use of Web 2.0 and other virtual technologies, applications are provided to users via the net with the data stored on the provider's servers

2. RELEATED WORK.

2.1 Farsite

Farsite is a server less, distributed file system that does not assume mutual trust among the client computers on which it runs [4]. Logically, the system functions as a central file server, but physically, there is no central server machine. Instead, a group of desktop client computers collaboratively establish a virtual file server that can be accessed by any of the clients. The system provides a global name space for files, location-transparent access to both private files and shared public files, and improved reliability relative to storing files on a desktop workstation. It does this by distributing multiple encrypted replicas of each file among a set of client machines. Files are referenced through a hierarchical directory structure that is maintained by a distributed directory service

2.2 Decentralized Erasure Code

The decentralized erasure codes which are randomized linear codes with specific probabilistic structure that leads to optimally sparse generator matrices. These codes canbe created by randomized network protocol. The decentralized erasure codes will reduce the communication, storage and computation cost over random linear coding .The decentralized erasure codes are the key factor to avoid the problem of large number of storage nodes with limited number of memory nodes. These codes will pre-rouets its data packet to O(log n) randomly and independently selected storage nodes. Decentralized erasure codes have minimal data node degree which corresponds to a maximal sparsity of the generator matrix and minimal number of pre routed packets

The key advantage of decentralized erasure code is that there is no need for co-ordination among the data nodes[5][7][10].

2.3 Proxy Re-Encryption

Proxy Re-Encryption scheme in which the proxy server can transfer a cipher text under a public key PKA to a new one under another public key PKB by using the re-encryption key RKA→B The server does not know the plain text during the transformation. The proxy Re- Encryption schemes gives better secured sharing methodology to the cloud server .When the user want to share the message to them neighborhood, user will send a re-encryption key to the storage server will do the re-encryption for the authorized user. This scenario describes the better confidentiality in the forwarding scheme.

2.4 Secure Cloud Storage Scheme

The cloud server will be having two type of storage servers. They are data server, key server. The data server which will be used to store the data packets after the decomposition of encrypted packets from the data owner The key server which is used to store keys for encrypted data. A decentralized in the cloud server gives better scalability because there is no central authority to control the storage server to join or leave. Making replicas of each message and store them in different server will improve the robustness of the cloud server.

2.5 Ocean Store

Ocean store system in composed of a multitude of highly connected pools among which data is allowed to flow freely. Clients connect to one or more pool, perhaps intermittently. An ocean store provide persistent about data in server the cloud. This architecture will provide continuous access to persistent information. Ocean store have two type of access control .They are ‗reader restriction' and ‗writer restriction'. Ocean store can be used to create very large digital libraries and repository for scientific data. Ocean store provides a common mechanism for storing and managing large data collections

3. PROPOED SYSTEM

In existing system, the data owner will forward his encrypted message to the user who made the request for the data, along with the user id to the cloud server. The storage server verifies the user- id for authentication purpose and allow the data owner to send the encrypted data packets to the new user. For the decryption process the combination of data owner's secret key and new user's public key. That key will be used for decryption process. By improving the data confidentiality we proposes sending the combination of secret key and the public key pair by sms. By doing this type of protection over transmitting of pair of keys will improve tight security concerns.

Architecture of Cloud Server Environment

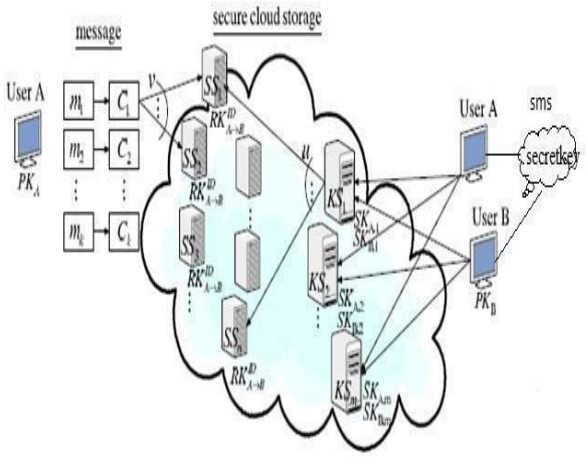

Fig 1 Architecture Diagram of Cloud Server Environment

m_1 , m_2 ..... m_n – Message blocks
c_1 ,c_2.. .c_n—Cipher text s of messages
SS — Storage Servers
KS – Key Servers
PK—Public Key
PR—Private Key

The above figure demonstrates the fundamental concept of our proposed work. There are five phases in our proposal. The first phase is the system management phase, in this phase the cloud manager chooses parameter of the cloud environment and publishes them. In our cloud environment there are ‗s' storage servers and ‗p' key servers are available. The storage servers which are used to store the data units of the data owner who upload the data. The key servers which are used to store key values of the each encrypted packets in the storage servers.

The second phase is the data Storage phase. In this phase User A encrypts his message using any encryption algorithm and send the encrypted texts to the cloud server. The cloud server will decompose them into multiple packets of data. These encrypted texts again will go for the encryption for improving the security mechanism, for both encryptions different type of keys will be used. After the completion process of encryption the cipher texts will store randomly chosen storage servers and the key for encrypted data will be maintained separately in the key servers[3]. Upon receiving cipher texts from a user ,each storage server linearly combines codeword symbols and store it.

The third phase is the data forwarding phase .User in the cloud server may want to share his data to another user in the cloud server or they want send the information to the outside user apart from cloud environment. In such cases the original user who want to share or forward his message ,first the user have to get an permission from the cloud server for an authorized entry for data manipulation in the cloud. For getting authorized entry, the user should have to append password as an verification code. After successful verification the user has the privilege to forward or modify the data.If the user forwarding the data means the receiver will receive the decrypted code only,do the decryption in receiving side require valid key.

The pair of keys will be made very confidentially by the cloud server, that decryption key containing the secret key of user and the public key of the receiver. The receiver‗s public key is known by the user who send the data from cloud ,how means the receiver while sending an request with public key to the user, he will notify the public key very secretly.

The final phase is the data retrieving phase, the user requests to retrieve a message from the cloud, the storage server holding data in cloud[10] .The message may be either stored by him in the cloud or forwarded to him in the cloud. Upon receiving the retrieval request from the cloud user and executing the proper authentication with the user, each key servers requests randomly chosen storage server to get the code word symbols and does the partial on the received codeword symbols by using the combined key

3.1 Data Owner

The data owner after registering with the cloud server, he will have the Admin password as a private key. With the help of that key he has the privilege to data access the information stored in the cloud server. Here the user who stores his information in the cloud server are termed as Data owner. He has the right to modify the existing data and he has the privilege to add the new data in the cloud server.

Before entering into the cloud sever, the data owner have to register them with the cloud server for the authorization purpose .Once the data owner has completed his registration, the cloud server will give the password for confidential uses of data manipulation purposes. That key should be maintained secretly ,otherwise it may bring out to wrong type of security

3.2 Cloud Server

The data owner stores the information in the cloud server, where the data is splitted into multiple packets randomly and thus each of the packets are stored in a separate cloud servers, namely the first packet to be stored in the cloud server(CS1),the second packet to be stored in the cloud server(CS2) and so on. Under the cloud server, there are both data servers and the key servers.

3.3 Randomized Network Algorithm

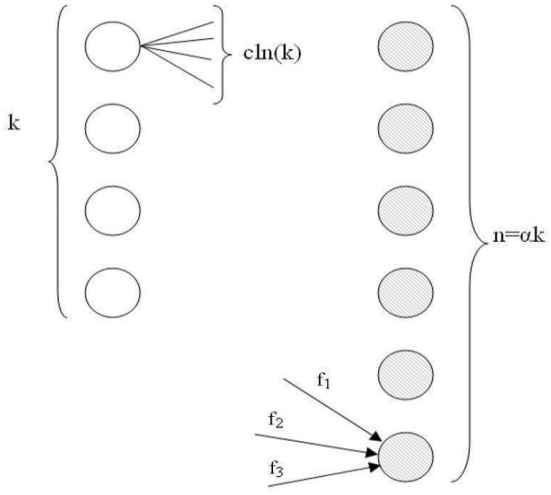

Fig 2 Randomized network algorithm

The Randomized Network algorithm is used to construct a decentralized erasure code in a cloud network. Each data node picks one out of the n storage nodes randomly ,pre-routes its packet and repeats d(k)= c ln (k) times .Each storage node multiplies whatever it happen to receive with coefficients selected uniformly and independently in Fq and stores the result and the coefficients

3.4 Encryption

In this proposal we use encryption scheme twice in the system, for improving security and Confidentiality in the cloud server. First encryption will be done before splitting the message into multiple packets and second encryption will be done for the splitted packets and those packets will be stored randomly chosen data servers. In each of the data server, encryption is done and thus the encrypted key will be generated for each of the data server[5],[7]

The first encryption process takes place before splitting data into smaller data units .After the completion of first encryption process the cloud server will format the encrypted data units in smaller number of equal cipher text units. These first level cipher texts get through one more time for encryption. In second level encryption, the cipher texts takes one more encryption process. This double encryption scheme provides higher level of tight security mechanism and increased data confidentiality for the data owners in the cloud environment.

Even though we are increasing the security mechanism, the double encryption scheme lead the cloud environment into performance degrading why because is that double encryption increase the execution time of the cloud environment.

3.5 SMS Alert

During the data sharing process, the user who needs the data get the encrypted keys from the data owner by means of the SMS [8],[9]. The keys are later utilized by the user in cloud servers and as a result he can get the data from the cloud servers which he needs. In order to improve the security and confidentiality of the combined key from the data owner to new user who got the encrypted data, that key send by an sms. By sending sms the intruder or the hacker may not aware of mobile number of the new user.

3.6 Error Recovery

If the storage server get fails due to any reasons suddenly, then the cloud will add the new storage for replace the failed one[6]. The new storage server quires the available storage servers, linearly combines the received Codeword symbols as a new one and store it back all data by using the erasure code mechanism.

4. CONCLUSIONS

This paper gives one of the approach to improve the security mechanism in data forwarding scheme in cloud server.This approach gives better confidentiality and provide good security rather than existing system. The future work can go with the direction by improving the scalability of the cloud and increasing the execution process of the cloud server.

ACKNOWLEDGMENTS

We thank our HOD and staff members of SRM University for guiding us throughout our research.

REFERENCES

[1] J. Kubiatowicz, D. Bindel, Y. Chen, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao,―Oceanstore: An architecture for global-scale persistent storage,‖ in Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS, pp. 190–201, ACM, 2000.

[2] P. Druschel and A. Rowstron, ―PAST: A large-scale, persistent peer-topeer storage utility,‖ in Proceedings of the 8th Workshop on Hot Topics in Operating System - HotOS VIII, pp. 75–80, USENIX, 2001.

[3] H.Ying Lin,Wen-Guey Tzeng‖A Secure Erasure code based cloud storage system with secure data forwarding‖ in Proceedings of the 8th Workshop on Hot Topics in cryptography – Hot CRY VIII, pp. 75–80, USENIX, 2011

[4] A. Adya, W. J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J. R. Douceur, J. Howell, J. R. Lorch, M. Theimer, and R. Wattenhofer, ―Farsite: Federated, available, and reliable storage for an incompletely trusted environment,‖ in Proceedings of the 5th Symposium on Operating System Design and Implementation - OSDI, pp. 1–14, 2002.

[5] A. Haeberlen, A. Mislove, and P. Druschel, ―Glacier: Highly durable, decentralized storage despite massive correlated failures,‖ in Proceedings of the 2nd Symposium on Networked Systems Design and Implementation - NSDI, pp. 143–158, USENIX, 2005.

[6] Z. Wilcox-O'Hearn and B. Warner, ―Tahoe: the least-authority filesystem,‖ in Proceedings of the 4th ACM International Workshop on Storage Security and Survivability - StorageSS, pp. 21–26, ACM, 2008.

[7] H.-Y. Lin and W.-G. Tzeng, ―A secure decentralized erasure code for distributed network storage,‖ IEEEnsactions on Parallel and Distributed Systems, vol. 21, pp. 1586–1594, 2010.

[8] D. R. Brownbridge, L. F. Marshall, and B. Randell, ―The Newcastle connection or unixes of the world

unite!,‖ Software Practice and Experience, vol. 12, no. 12, pp. 1147–1162, 1982.

[9] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon, ―Design and implementation of the sun network filesystem,‖ 1985.

[10] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, ―Plutus: Scalable secure file sharing on untrusted storage,‖ in Proceedings of the 2nd USENIX Conference on File and Storage Technologies - FAST, pp. 29–42, USENIX, 2003.