International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 02 Issue: 09 | Dec-2015 www.irjet.net p-ISSN: 2395-0072

Secure Way to Data Storage and Forwarding Using Cloud Computing

Miss.Vasanti K. More[1], Mr.Kaushik H. Udavant[2], Mr.Vaibhav V. Tandale[3], Mr.Gaurav P. Dixit[4]

[1] *Sir Visvesvaraya Institute Of Technology, Department of Computer Engineering, Savitribai Phule Pune University Nashik, India*
[2] *Sir Visvesvaraya Institute Of Technology, Department of Computer Engineering, Savitribai Phule Pune University Nashik, India*
[3] *Sir Visvesvaraya Institute Of Technology, Department of Computer Engineering, Savitribai Phule Pune University Nashik, India*
[4] *Sir Visvesvaraya Institute Of Technology, Department of Computer Engineering, Savitribai Phule Pune University Nashik, India*

Abstract - *Cloud storage is a technique of networked online storage where data is store in multiple servers. Organization cite data confidentiality is their serious concern for cloud computing, with un-encoded data stored on another system server of cloud system.*

We propose a new encryption scheme and integrate it with a secure decentralize code to form secure data storage distributed system. The encryption scheme support encoding operation and over encrypted message and resending operation over encrypted and encoded information. The encrypted data is split and store in multiple server of cloud. This split data is merge at the time of download data by user. The rigid integration of encoding and encryption, and forwarding creates storage system efficiently meets the requirement of data healthiness, data privacy, data forwarding. Accomplishing the assimilation consideration of distributed structure is performing. Our system meets the requirement that storage servers independently perform encoding and re-encryption and key servers independently to perform complete decryption.

Key Words: *Distributed Data Storage, Identify-based System, Encryption , Decryption, Security.*

1.INTRODUCTION

CLOUD computing provides users with a convenient mechanism to manage their personal file with the notion called database-as-a-service (DAS) . In DAS schemes, user can outsource his encrypted files to untrusted proxy servers. Proxy servers can perform some functions on the outsourced ciphertexts without knowing anything about the original files. Unfortunately, this technique has not been employed extensively. The main reason lies in that users are especially concerned on the confidentiality, integrity and query of the outsourced files as cloud computing is a lot more complicated than the local data storage systems, as the cloud is managed by an untrusted third party. After outsorcing the files to proxy servers, the user will remove them from his local machine. Therefore, how to guarantee the outsoured files are not accessed by the unauthorized users and not modified by proxy servers is an important problem that has been considered in the data storage research community.
Furthermore, how to guarantee that an authorized user can query the outsourced files from proxy servers is another concern as the proxy server only maintains the outsourced ciphertexts. Consequently, research around these topics grows significantly.

Confidentiality is proposed to prevent unauthorized users from accessing the sensitive data as it is subject to unauthorized disclose and access after being outsourced. Since the introduction of DAS, the confidentiality of outsourced data has been the primary focus among the research community. To provide confidentiality to the outsourced data, encryption schemes are deployed .

Integrity can prevent outsourced data from being re-placed and modified. Some schemes have been proposed to protect the integrity of the outsourced data, such as proof of retrievability and prov-able data possession . In these schemes, digital signature schemes and message authentication codes (MAC) are deployed.

Query data storage is executed between a receiver and proxy server. The proxy server can perform some functions on the outsourced ciphertexts and convert them to those for receiver. As a result, the receiver can obtain the data outsourced by the owner without the proxy server knowing the content of the data.

2.OBJECTIVE

In this sector, we review schemes associated to identity-based

secure distributed records storage (IBSDDS) schemes.

1.1.1 Data Storage Systems

Outsourcing expanding from the data confidentiality to data utility, and pointed out the main research directions in the protection of the externally stored data. Kher and Kim surveyed the data storage systems com-prehensively and classified them into three kinds based on their security services: networked file systems (NFS), storage-based intrusion detection systems (SBIDS) and cryptographic file systems (CFS).

Networked File Systems. In these systems, proxy servers are assumed to be trusted. They authenticate receivers and validate access permissions. The interactions linking the proxy servers and receivers are executed in a secure channel. Therefore, these systems cannot provide an end-to-end data security, namely they cannot ensure the confidentiality of the data stored at the proxy server . In these schemes, a receiver authenticates himself to the proxy server using his password. Then, the proxy sever passes the authentication result to the file owner. The owner will make an access permission according to the received information.

Storage-based interruption Detection Systems. In these sys-tems, an intrusion detection scheme is embedded in proxy servers or the file vendor to detect the intruder's behaviors, such as adding backdoors, inserting Trojan horses and tampering with audit logs. These schemes can be classified into two types: host-based system and network-based system. In the host-based systems, an intrusion detection scheme is embedded in the host to detect the local intrusion actions. On the contrary, in network-based systems, an intrusion detection scheme is embedded in the proxy servers to detect the external intruder's actions. The major advantage of these systems is that proxy servers can still detect the intrusion events even if the host is compromised as the proxy server are independent from the host.

Cryptographic File System. In these systems, an end-to-end security is provided by cryptographic protocols which are executed by the file owner to prevent proxy servers and illegal users from modifying and accessing the sensitive files. These systems can be di-vided into two types: shared file system and non-shared system. In shared file systems , the vendor can share his files with a group of users. Cryptographic techniques deployed in these systems are key sharing, key agreement and key revocation. In non-shared file systems , in order to share a file with another user, the owner can compute an access key for the user using his secret key. In these two systems, the reliability of the responsive files is provided by digital signature schemes and message authentication codes (MAC).

3.LITERATURE SURVEY

We consider the problem of constructing an erasure code for storage over a network when the data sources are distributed. Specifically, we assume that there are n storage nodes with limited memory and k < n sources generating the data. We want a data collector, who can appear anywhere in the network, to query *any* k *storage nodes* and be able to retrieve the data. We introduce Decentralized Erasure Codes, which are linear codes with a specific randomized structure inspired by network coding on random bipartite graphs. We show that decentralized erasure codes are optimally sparse, and lead to reduced communication, storage and computation cost over random linear coding.

Plutus is a cryptographic storage system that enables secure file sharing without insertion much trust on the file servers. In particular, it makes novel use of cryptographic primitives to protect and share files. Plutus features highly scalable key management while allowing individual users to retain direct control over who gets access to their files. We explain the mechanisms in Plutus to reduce the number of cryptographic keys exchanged between users by using filegroups, distinguish file read and write access, handle user revocation efficiently, and allow an untrusted server to authorize file writes. We have built a prototype of Plutus on OpenAFS. Measurements of this prototype show that Plutus achieves strong security with overhead comparable to systems that encrypt all network traffic.

4.Architecture Diagram

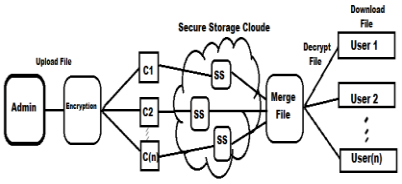

Fig1:Data send And receive Architecture

In Admin Module the admin can login to provide his username and password. Then the server setup method

can be opened. In server setup process the admin _rst set the remote servers Ip-address for send that Ip-address to the receiver. Then the server can skip the process to activate or Dis- activate the process. For activating the process the storage server can display the Ip-address. For Dis-activating the process the storage server cannot display the Ip-address. These information can be viewed by clicking the key server. The activated Ip-addresses are stored in available storage server. By clicking the available storage server button we can vision the currently available Ip-addresses.

3. CONCLUSIONS AND FUTURE WORK

Adaptive Encryption Scheme supports programming and forwarding, and influenced decryption process in a sending way. To decrypt a message of k parts that are encrypted and encoded to n secret code symbols, every key server only has to in various measure decrypt two codeword symbols into our system. By using the threshold Adaptive Encryption Scheme, we consider a safe cloud storage system that provides secure data storage and secure data forwarding functionality in a decentralized structure. Moreover, all storage server in matching performs encoding and reencryption and each key server independently execute partial decryption.

REFERENCES

1. H. Hacig¨um¨us, B. R. Iyer, C. Li, and S. Mehrotra, “Executing SQL over encrypted data in the database-service-provider model,” in *Proceedings: SIGMOD Conference - SIGMOD’02* (M. J. Franklin, B. Moon, and A. Ailamaki, eds.), vol. 2002, (Madison, Wisconsin, USA), pp. 216–227, ACM, Jun. 2014.
2. L. Bouganim and P. Pucheral, “Chip-secured data access: Confidential data on untrusted servers,” in *Proc. International Conference on Very Large Data Bases - VLDB’02*, (Hong Kong, China), pp. 131–142, Morgan Kaufmann, Aug. 2002.