INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

CDINS: Cooperative Provable Data Possession for Integrity Verification in Network-Coding-Based Multi Cloud Storage System

Mohan Kumar G.
M.Tech. (PG Scholar), Department of CSE, RRCE, Bengaluru, India
Dimpu Sagar N.
M.Tech. (PG Scholar), Dept of CSE, RRCE, Bengaluru, India
Aakanksh S. G.
Software Engineer, Auro Display (India) Pvt. Ltd., Bengaluru, India

Abstract:
Cloud Computing has improved in its existence, when compared to grid computing and cluster computing by providing an optimized and economical solution for sharing group resources among the cloud users. Persistent data, maintaining privacy in an untrusted cloud is still a demanding issue. In this paper we propose a secure owner, data sharing using advance encryption standard (AES), and splitting the data using remote file synchronization single round algorithm (RSYNC), data migration between clouds is an important feature. Here we use proxy based storage for fault tolerance in multiple-cloud storage called network coding, which overcomes, problems such as loss of data, permanent failure. In our scheme we represent five cloud servers, four cloud servers provide the requested data to end-user, if the data is hacked or deleted, the storage regenerating code (SR) retrieves the data from the fifth cloud server and delivers to end-user. A graph is plotted for the total time taken for the regenerating code. The main feature of SR code is that we release the encoding requirement of storage nodes during repair, to make regenerating code portable for any cloud storage. In addition, we analyze the security of our scheme with rigorous proofs and demonstration the efficiency of our scheme in experiments.

Keywords: *Cloud Computing, Cloud Server, Data Sharing, Experiment, Implementation, Storage Regeneration Code*

1. Introduction

The demanding growth of high speed network in accessing internet for industrial, commercial and entertainment purpose comprised of thousands of concurrent ecommerce transaction in every day. To handle this demand to access high speed network it needs large scale data-center, thousands of servers, large infrastructure.[3] So many organizations such as Google, eBay, salesforce.com, HP, IBM, operating huge data center to whole world. Many organizations invest huge amount of capital, time, infrastructure, large data center. One of the most fundamental services offered by cloud providers is data storage. By making use of Scloud [3], the work can be completely freed from the troublesome local data storage and maintenance.

Cloud computing provides popular on-line services for archiving, backup, and even primary storage of files, and transforming business by offering new options for businesses to increase efficiencies while reducing costs [2]. It lets user can access all applications and documents from anywhere in the world, freeing from the confines of the desktop and making it easier for group members in different locations to collaborate. However, it also poses a significant risk to the confidentiality of those stored files. The cloud service provider or third party may not fully trusted by the users. While the data files stored in the cloud may be sensitive and confidential. The basic solution is to encrypt the data and then upload the encrypted data into the cloud.

Proxy servers works as storage system and is designed for providing fault-tolerant, repair, by providing data if it is lost, corrupted by regenerating the data with he help of storage regenerating code (SR) through network coding (NC) which can interconnect different clouds and strip data across the clouds[1].

The draw backs of the existing system can be overcome using this proposed system [1]. The main contribution of this paper is that : The proposed system use advance encryption standard (AES), where the data provider encrypts the files before uploading, once uploaded we use remote file synchronization single round algorithm (RSYNC)[4] for splitting the encrypted files, these chunks of files are further digested by message authentication code (MAC) and uses Sha-1 algorithm for generating accurate key for each chunk which provides meta data and local data that is uploaded to the proxy server. The important feature in this scheme is data migration, which stores the uploaded data between cloud servers, for retrieval of data if it is lost or corrupted.

2. Architecture

The architecture consists of different entities as illustrated in the below figure. Data Provider (owner), Cloud Servers, Proxy Server, Data Consumer (End User)

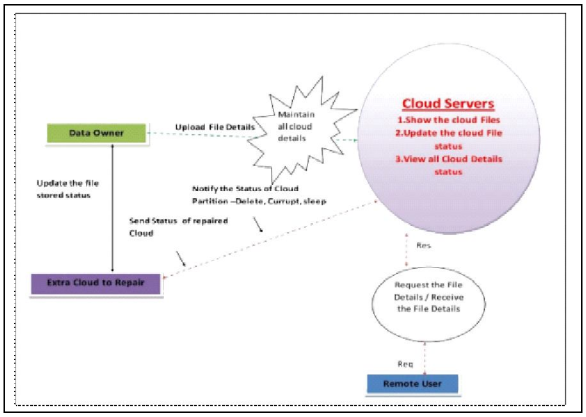

Data Owner
Upload File Details
Maintain all cloud details
Cloud Servers
1.Show the cloud Files
2.Update the cloud File status
3.View all Cloud Details status
Update the file stored status
Notify the Status of Cloud Partition –Delete, Currupt, sleep
Send Status of repaired Cloud
Extra Cloud to Repair
Res
Request the File Details / Receive the File Details
Req
Remote User

Figure 1: Architecture

2.1. Data Provider (Owner)

In this module, the data owner uploads the data in the cloud server. For the security purpose the data owner encrypt the data file and then store in the cloud. The Data owner is capable of manipulating the encrypted data file. And the data owner can set access privilege to the encrypted data file.

2.2. Proxy storage system (Proxy Server)

Proxy server is an interface between cloud servers and data provider, the main function is to interconnect multiple cloud repositories, and to avoid session logs. Trust worthiness will be established between cloud servers and data provider, network coding deals with read/ write requests, and interconnect different clouds and strip data across the cloud servers.

2.3. Cloud Server

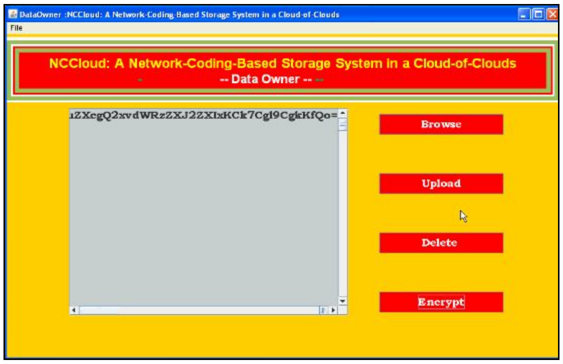
All the uploaded file will be currently in cloud servers via the proxy server, it manages cloud to provide data storage service between following servers (cs1, cs2, cs3 and cs4). The consumer access the requested data by decrypting the data files.

3. Implementation Work

3.1. Data Provider

The data provider uploads the file to the proxy server, for security purpose files are encrypted using advance encryption standard (AES), once uploaded we use remote file synchronization single round algorithm (RSYNC)[4] for splitting the encrypted files, these chunks of files are further digested by message authentication code (MAC) and uses Sha-1 algorithm for generating accurate key for each chunk which provides meta data and local data that is uploaded to the proxy server, the snap shot represents the various tasks handled by data owner Some of the task of the data provider is that:

- Upload a file
- Data Confidentiality
- Migration


DataOwner :NCCloud: A Network-Coding-Based Storage System in a Cloud-of-Clouds
File
NCCloud: A Network-Coding-Based Storage System in a Cloud-of-Clouds
-- Data Owner --
Browse
Upload
Delete
Encrypt

1ZXcgQ2xvdWRzZXJ2ZXIxKCk7Cgl9CgkKfQo=

Figure 2: Data Provider

3.1.1. Upload a File

The uploaded file will be in the encrypted format, by using AES algorithm, further these files are divided into chunks, later we encoding coefficient vector(ECV) which contain metadata i.e. native chunks. The code chunks are then evenly stored in the storage nodes. The uploaded file can be downloaded or decrypted by using a secret key, which will be generated for each and every file.

3.1.2. Data Confidentiality

Data confidentiality requires that unauthorized users including the cloud are incapable of learning the content of the stored data. An important and challenging issue for data confidentiality is to maintain its availability for dynamic cloud servers.

3.1.3. Migration

The important feature in this scheme is data migration, which stores the uploaded data between cloud servers, for retrieval of data if it is lost or corrupted. However, the practical performance of regenerating codes remains uncertain.

3.2 Cloud Server

Cloud Server receives the chunks of file in encrypted format via proxy server, reads the essential data pieces from other surviving clouds as shown in the snap shots, reconstructs new data pieces, and writes these new pieces to a new cloud. The file system layer is present as a mounted drive on NC Cloud which can thus, be easily interfaced with general user applications. The coding layer deals with the encoding and decoding functions. To access the shared data files, data consumers download encrypted data files of their interest from the cloud and then decrypt them.

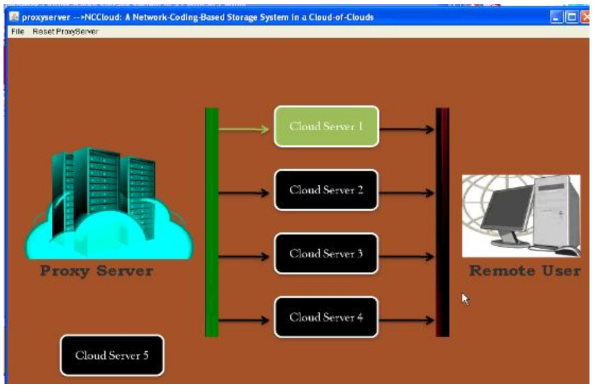
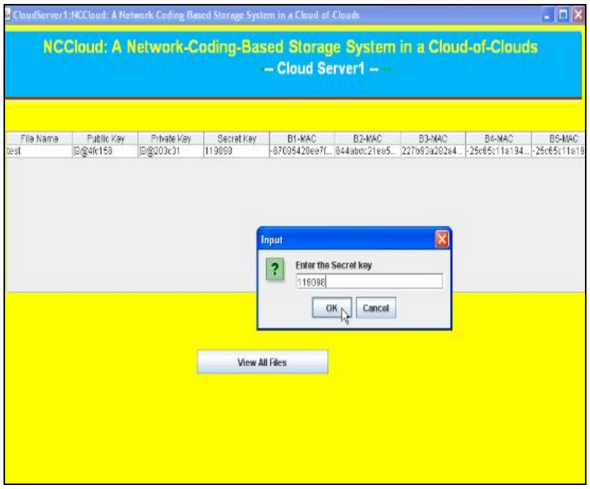


Figure 3: Proxy Server


CloudServer1:NCCloud: A Network Coding Based Storage System in a Cloud of Clouds

NCCloud: A Network-Coding-Based Storage System in a Cloud-of-Clouds
-- Cloud Server1 --

File Name	Public Key	Private Key	Secret Key	B1-MAC	B2-MAC	B3-MAC	B4-MAC	B5-MAC
test	[B@4fc156	[B@203c31	119099	-87095420ee7(...	844abdc21ee5...	227b93a282a4...	-25c65c11a194...	-25c65c11a19

Input

Enter the Secret key

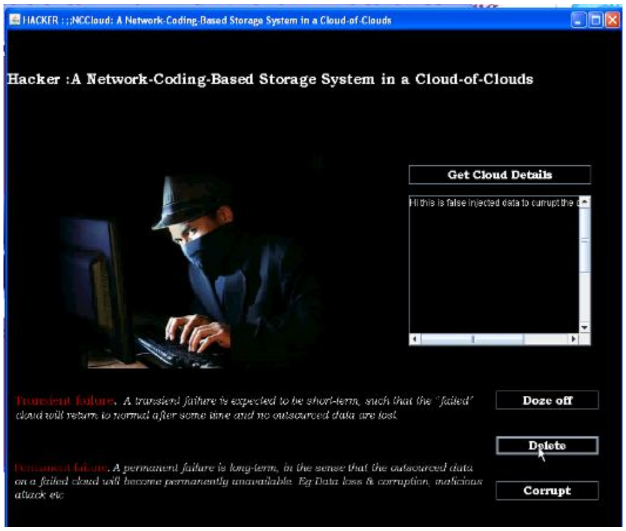
119098

Figure 4: Cloud Server

3.3 Hacker

Attacker can attempt to temporary failure for a cloud by making Doze Off for a particular period of time. He can also attempt to permanent failure by Deleting and corrupting the cloud. Then the unauthorized user will considered as an attacker as shown in the snapshot. The different methods implemented in this paper is:

- Deleting the data
- Corrupting the file
- Doze Off (Delay in time)



HACKER : ;;NCCloud: A Network-Coding-Based Storage System in a Cloud-of-Clouds

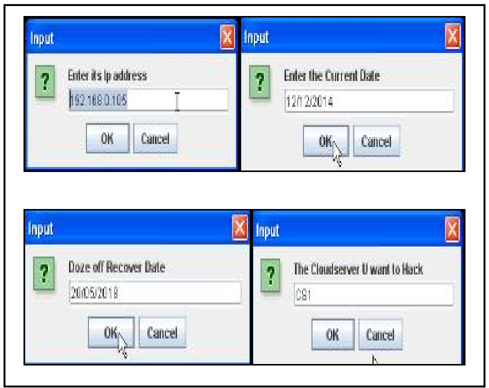
Hacker :A Network-Coding-Based Storage System in a Cloud-of-Clouds

Hi this is false injected data to currupt the d

Transient failure. A transient failure is expected to be short-term, such that the 'failed' cloud will return to normal after some time and no outsourced data are lost.

Permanent failure. A permanent failure is long-term, in the sense that the outsourced data on a failed cloud will become permanently unavailable. Eg Data loss & corruption, malicious attack etc

Figure 5: Hacker



Input

Enter its Ip address: 192.168.0.105

Input

Enter the Current Date: 12/12/2014

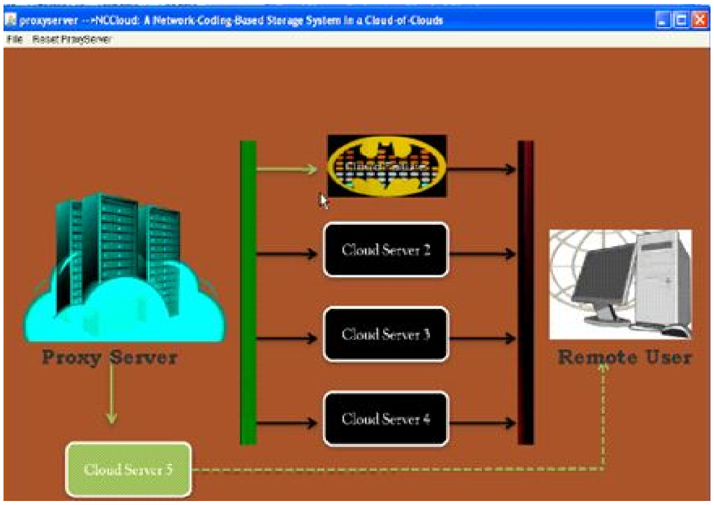
Input

Doze off Recover Date: 20/05/2018

Input

The Cloudserver U want to Hack: CS1

Figure 6: Attacked Cloud



proxyserver -->NCCloud: A Network-Coding-Based Storage System in a Cloud-of-Clouds

Figure 7: Recovery Cloud Server

3.4 Data Consumer (End user)

In this module, the user can only access the data file with the encrypted key if the user has the privilege to access the file. For the user level, all the privileges are given by the Data provider authority. Data can be accessed by users within the cloud solitary. If the requested file is deleted or corrupted the storage regeneration code fetches the requested data and gives it to the extra cloud server i.e. Cloud server 5, which ensures the end user by the requested data without any delay.

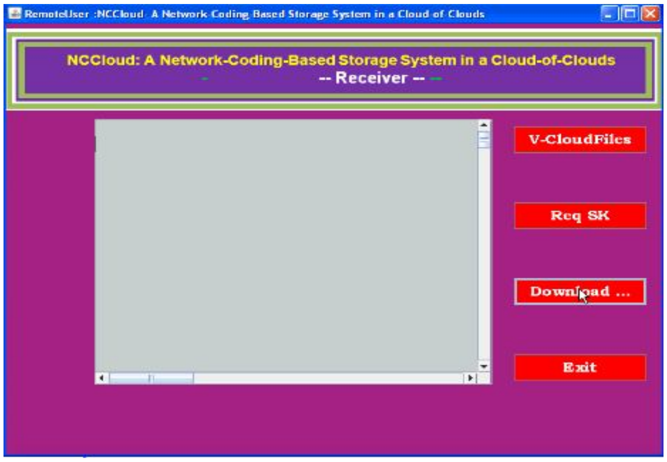

RemoteUser :NCCloud: A Network Coding Based Storage System in a Cloud of Clouds
NCCloud: A Network-Coding-Based Storage System in a Cloud-of-Clouds
-- Receiver --
V-CloudFiles
Req SK
Download ...
Exit

Figure 8: End User

4. Results Discussion

The below graph represents the number of revoked users present in each and every group, the revoked user in the group1 are represented in red. The revoked users in the group2 are represented in blue whereas the revoked users in group3 are represented in green as seen in the graph. The numbers of revoked users in group1 are more when compared to the group 2 and group3.

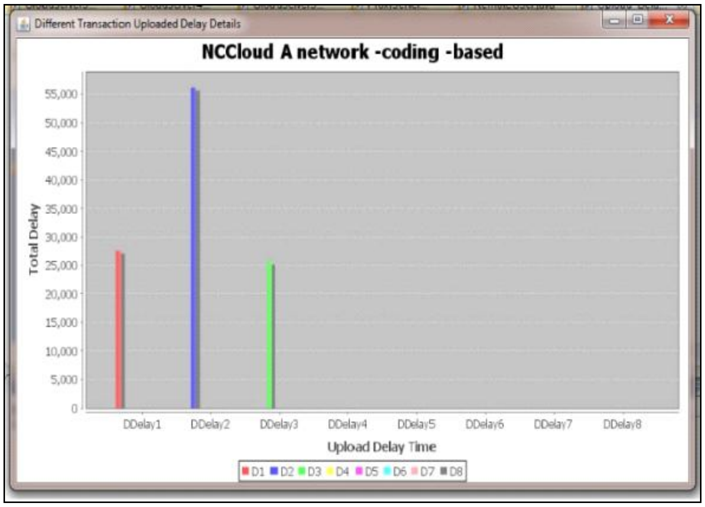


Figure 9: The total time delay for uploading the file

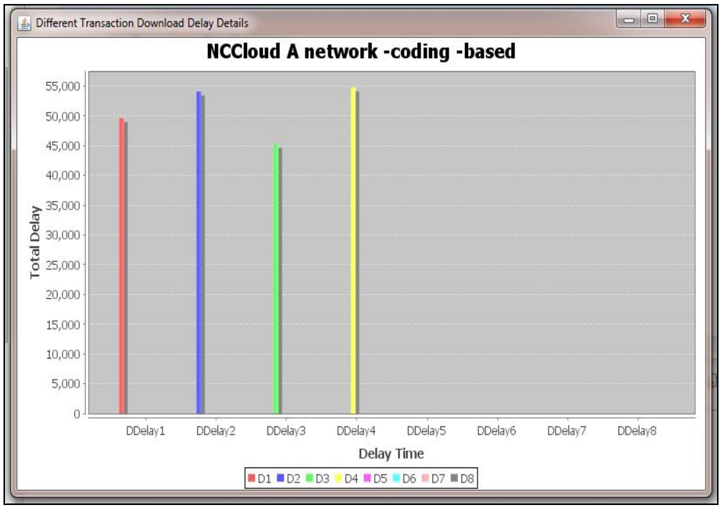


Figure 10: The total time delay for downloading the file

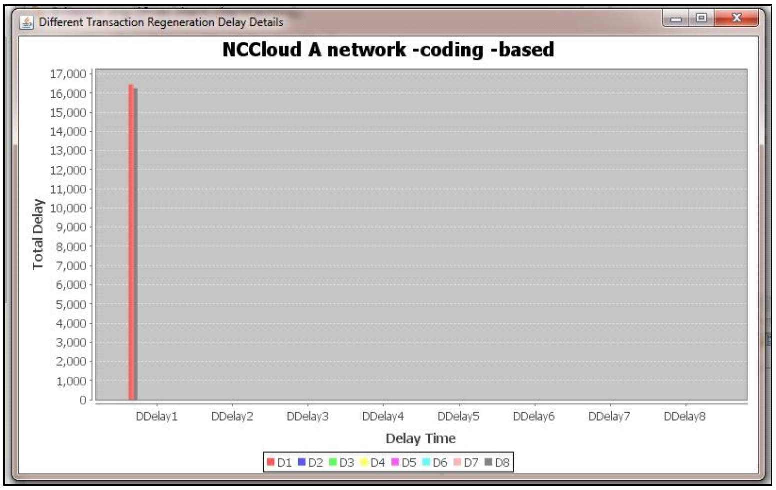


Figure 11: The total time for Regenerating Delay Details

5. Conclusion

In this paper we design a secure data sharing scheme and achieves data confidentiality, and hoop data across multiple cloud vendors. We use storage regeneration code if the file is deleted or corrupted by an attacker, for retrieving the lost data. Moreover, the storage overhead and the encryption computation cost are constant. Extensive analyses show that our proposed scheme satisfies the desired security requirements and guarantees efficiency as well.

6. References

i. Mohan Kumar G , Neelu L, "RRCE: Secure and Dependable Network Coding for the Storage Repair in a Cloud-Of-Clouds " On cloud computing on 26th April 2015.
ii. M. Armbrust, A. Fox, R. Griffith, A.D. Joseph R.H Katz, A. Konwinski, G. Lee, A.D Patterson, A. Rabkin, I Stoica, and M. Zaharia “A View of Cloud Computing,” comm.ACM vol. 53, no. 4, April 2010.
iii. M. Armbrust, A. Fox, R. Griffith, A.D. Joseph R.H Katz,A. Konwinski, G. Lee, A.D Patterson, A. Rabkin, I Stoica, and M. Zaharia “ A View of Cloud Computing,” comm.ACM vol. 53, no. 4, pp. 50-58, April 2010
iv. A. Tridgell and P. Mackerras. The rsync algorithm. Technical Report TR-CS-96-05, Australian National University, June1996.