Dynamic Resource Allocation and Data Security for Cloud

Rajesh M. Devakate
Annasaheb Dange College of Engg. & Technology,
Ashta, Maharashtra, India.

Amol B. Rajmane
Ashokrao Mane Group of Institutions, Vathar Tarf
Vadgaon, Maharashtra, India.

Abstract*— Cloud computing is the next generation of IT organization. Cloud computing moves the software and databases to the large centres where the management of services and data may not be fully trusted. In this system, we focus on cloud data storage security, which has been an important aspect of quality of services. To ensure the correctness of user's data in the cloud, we propose an effective scheme with Advanced Encryption Standard and MD5 algorithm. Extensive security and performance analysis shows that the proposed scheme is highly efficient. In proposed work we have developed efficient parallel data processing in clouds and present our research project for parallel security. Parallel security is the data processing framework to explicitly exploit the dynamic storage along with data security. We have proposed a strong, formal model for data security on cloud and corruption detection.

***Keywords*— Cloud computing, Resource Allocation, cryptography Encryption, Decryption, corruption detection.**

I. INTRODUCTION

Several trends are opening in cloud computing, which is an Internet-based and use of computer technology to development. The cheaper and more powerful processors with the software as a service (SaaS) computing architecture are transforming data centres into pools of computing service on a large scale. The increasing network bandwidth and reliable network connections make it even possible that internet users can subscribe high quality services from data and software that resides on remote centres. Moving data into the cloud storage offers great benefit to users since they don't have to care about the difficulties of direct hardware management. The colonizer of cloud computing vendors, Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2) [1] are both known examples. While these online services provides large amounts of storage space and customizable computing resources, this platform shift, however, is eliminating the responsibility of local machines for data storage and maintenance at the same time. As a result, users are at the leniency of their cloud service providers for the availability and integrity of their data. From the perspective of data security, which has always been an important part of quality of service, cloud computing poses new challenging security threats for number of reasons. Such as traditional cryptographic system for the purpose of data security protection cannot be directly adopted due to the user's loss control of data under cloud storage. Therefore, verification of correct data stored in the cloud must be conducted without knowledge of the whole data. Considering various kinds of data for each user stored in the cloud and the demand of continuous assurance of their data safety, the problem of verifying correctness of data storage in the cloud becomes even more challenging. Cloud computing is not just a third party data warehouse. The data stored in the cloud may be updated by the users. However, this dynamic feature also makes traditional integrity insurance techniques. The deployment of cloud computing is powered by data centres running in a simultaneous, cooperated and distributed manner. Individual user's data is redundantly stored in multiple locations to further reduce the data integrity threats. Therefore, distributed protocols for storage correctness assurance will be of most importance in achieving a robust and secure cloud data storage system in the real world. However, such important area remains to be fully explored in the literature.

This techniques, while can be useful to ensure the storage correctness without having users possessing data and they are all focusing on single server scenario and most of them do not consider dynamic data operations. As a complementary approach, researchers have also proposed distributed protocols [8] for ensuring storage correctness across multiple servers. Again, none of these distributed schemes is aware of dynamic data operations. As a result, their applicability in cloud data storage can be drastically limited.

In this paper, we proposed an effective and flexible distributed scheme with explicit dynamic data support to ensure the correctness of user's data in the file distribution preparation to provide redundancies and guarantee the data dependability. This construction drastically reduces the communication and storage overhead as compared to the traditional replication-based file. To process/allocate jobs like customer documentation upload to cloud server, we will provide facility of admin to feed input to main server i.e. documents will be uploaded by admin on main server. Further our system will automatically process data in parallel feed to cloud nodes for storage. Obviously, these documents will be chunked in multiple parts and with the encryption. Once admin/user uploads desired documents, the basic working platform will be ready for actual research. The file/resources storage on cloud is common application of corporate cloud. But if any loss of data occurred due to virus attack or hacking, it will be difficult to recover.

Proposed work focuses on security and corruption detection of data. For this purpose we developed own system which co-actively work with parallel data processing approach. We also utilized concept of MD5 to tag files with document identity number. Hence if any part of file missing system can recover data using anti-parallel resource algorithm.

II. Literature review

Cloud computing is providing unlimited infrastructure to store and execute customer data and program. As customers you do not need to own the infrastructure, they are merely accessing or renting; they can forego capital expenditure and consume resources as a service, paying instead for what they use. Cloud storage can save users lots of money over local storage solutions [1]. When user uses the cloud, user doesn't know exactly where your data is hosted, what country it will be stored in? Data should be stored and processed only in specific jurisdictions as define by user. Provider should also make a contractual commitment to obey local privacy requirement on behalf of their customers. Data-centred policies that are generated when a user provides personal or sensitive information, that information travels throughout its lifetime to ensure that the information is used only in accordance with the policy [5]. Data that is generated during running of program on instances is all customer data and therefore provider should not perform backups [7, 8].

Cong Wang investigated the problem of data security [2] in cloud data storage, which is essentially a distributed storage system but doesn't provide recovery facility. The processing frameworks which are currently used have been designed for static, homogenous cluster setup and disregard the particular nature of cloud [3]. Fast and secure message authentication in cryptography [4] provides splitting of file only supports for encryption and decryption of file in cloud computing. Proofs of retrievability protocol represents the proof of data stored in server is intact and retrievable but not support for file updates, as well as publicly verifiable PORs.

III. Comments

As there is no facility of data recovery available in cloud domain, this is very crucial application to prove efficient method for missing data recovery. The cloud's virtualized nature helps to enable promising new use cases for efficient parallel data processing. However, it also imposes new challenges compared to classic cluster setups. The major challenge we see is the cloud's opaqueness with prospect to exploiting data locality. For security reasons clouds often incorporate network virtualization techniques which can hamper the inference process, in particular when based on latency measurements.

To provide efficient parallel data processing for resource allocation we need to consider greater security while resources being allocated, data loss prevention and efficient allocation of cloud storage resources. Many cloud infrastructures are facing problems in parallel processing of data. Such problems may encounter due to hacking methodologies.

IV. Need of Work

- To provide secure cloud storage for registered clients.
- To provide security against data hacking / server hacking
- To provide way for administrator to maintain user profiles and data profiles.
- To provide security for files through encryption concept.
- To provide multi-server facility for data storage i.e. to make multi-parts of client files.
- To allocate resources / files based upon storage capacity.
- To recover infected or corrupted resources /files.

V. Proposed work

To provide parallel data allocation and data security we have stored data on multiple cloud nodes and in case if any data/part of file is missing or virus infected then we can able to recover data/resources. For recovery purpose we have implemented self-derived algorithm which collectively work with MD5, dynamic data allocation approach and logical addresses of file. We aim to recover data 100% which is not possible with existing systems.

In our proposed system client can be able to register with cloud service provider. Registered client have to login to upload file which he/she want to store on cloud. Here client expects higher level of security for his/her resources. In this system server receives resources uploaded by client. Server reviews received resource to filter banned data/documentation. The main server receives resources uploaded by client. Consider user have upload "test.txt" file proposed algorithm encrypts the file and splits into number of data chunks. These splitted data chunks parallelly processed to store on auxiliary cloud server-1 and auxiliary cloud server-2 shown in figure 1. This system provides more security for data and there will not be any direct access to user to auxiliary server's data. We have implemented MD5 algorithm to check all files are stored as it is or any files are modified due to impact of any mishap. In case of data corruption or missing file part, we can recover original data from cloud server by using MD5 hash value.

A. *Client Registration Facility for New Clients & Login Facility for Existing Clients*

We have developed client registration and login window. This will register new user with system to avail system facility. In this module existing client can log-in to upload his/her data file. If user is registered then he/she gets ID & password. At the time of login user has to provide correct ID & password. Then system checks the user ID & password if both are correct then user logged into the system. Unauthorized person cannot be able to login into the system.

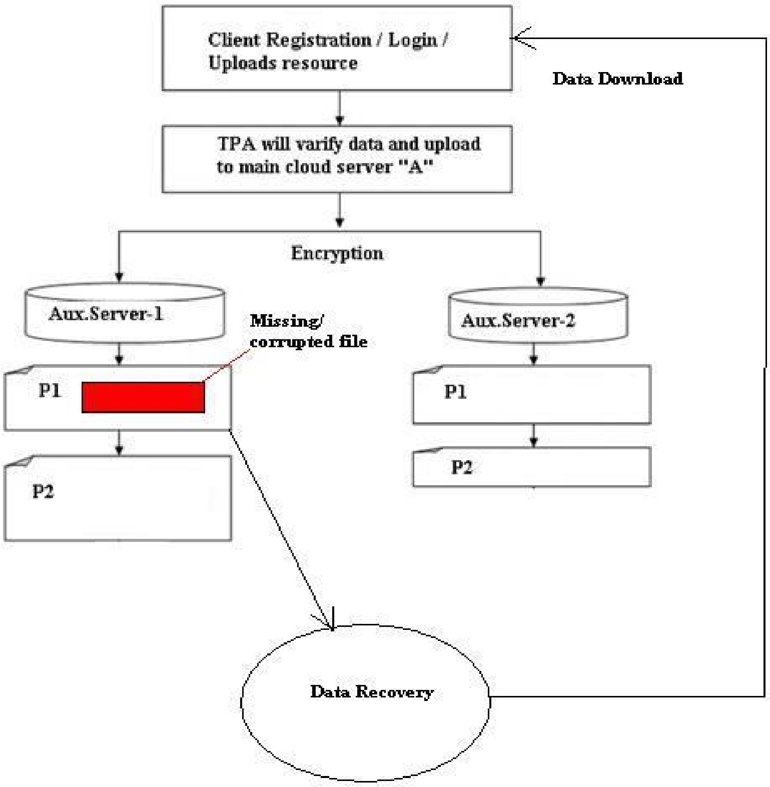

Client Registration / Login / Uploads resource
Data Download
TPA will varify data and upload to main cloud server "A"
Encryption
Aux.Server-1
Missing/ corrupted file
Aux.Server-2
P1
P2
P1
P2
Data Recovery

Figure 1: System Architecture

B. File Encryption and Upload

In this module, when user is registered he/she uploads file on main server and the main server encrypts the file and stored on server.

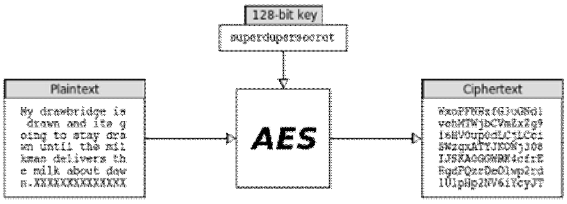


Figure 2: AES Encryption

The Advanced Encryption Standard (AES) algorithm is implemented for encryption of the data shown in figure 2. The main server receives files uploaded by client. Then file is splitted into number of data chunks. The AES algorithm will encrypt the splitted file parts. These data will be parallelly processed to store on nodes.

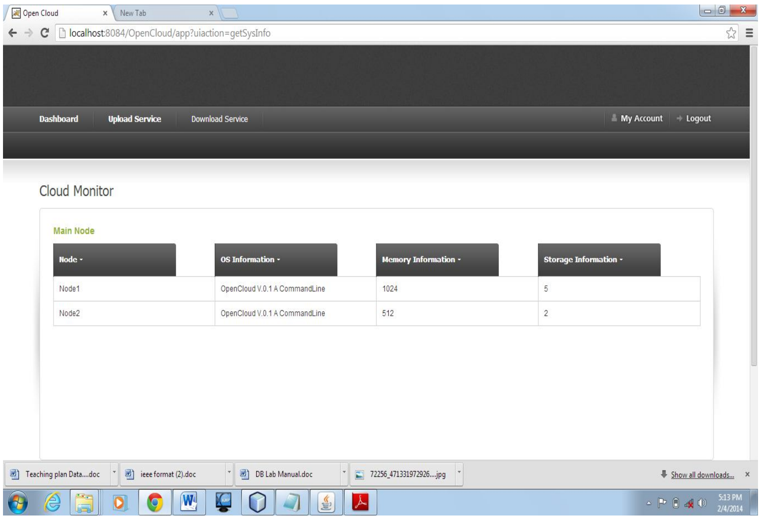
C. File Decryption and Download

In this module, user can download the files from server. The main server decrypts the file using AES algorithm and gives to the user. The AES algorithm is also implemented for decryption of the data. This will provide more security for data and there will not be any direct access of user to auxiliary server's data.

D. Dynamic Resource Allocation

In this module, we have created virtual nodes for allocating different servers for storage. In our project, we have created nodes for storing the files shown in figure 3. Suppose file is encrypted and stored on different nodes. When we are trying to download file then file parts are collected from various nodes.

We have presented a system that uses virtualization technology to store data resources (file) dynamically and support green computing by optimizing the number of servers in use. We have combined different types of workloads (storage) nicely and improved the overall utilization of server resources. We have developed a set of heuristics that prevent overload in the system effectively while saving energy used.



Cloud Monitor

Main Node

Node	OS Information	Memory Information	Storage Information
Node1	OpenCloud V.0.1 A CommandLine	1024	5
Node2	OpenCloud V.0.1 A CommandLine	512	2

Figure 3: Node Information

E. *File Corruption Detection and Recovery*

In this module, we have implemented MD5 algorithm. User's file is splitted and encrypted using AES algorithm. That file parts are stored on different nodes. By using MD5, we have calculated hash value and stored on database. When user wants to download his/her file then again hash value of current file is calculated and verified with old hash value. If both hash values are matched then user gets his original file. If hash values are not same then we can say that file parts are corrupted or infected. When we know that file is corrupted then we can recover this file.

VI. Results

TABLE I
File & Fragmentation size for 2 Node

	File size	**Fragment**
F1	50KB	25KB
F2	50MB	25MB
F3	500MB	250MB
F4	1GB	512MB

Table I shows the file size and fragment size. We are using two nodes for storage then file is splitted into two parts.

TABLE II
Comparison of Encryption Algorithms

File	**Average Encryption time (Milliseconds)**		
	AES128	**AES256**	**DES**
F1	0.00055	0.0008	0.00135
F2	0.0006	0.00095	0.00155
F3	0.0007	0.0012	0.0018
F4	0.00095	0.0014	0.0020

Table II shows the file and time required for encryption. We have taken three different algorithms for comparison viz. Advanced Encryption Standard (128bits), AES (256bits) and Data Encryption Standard.

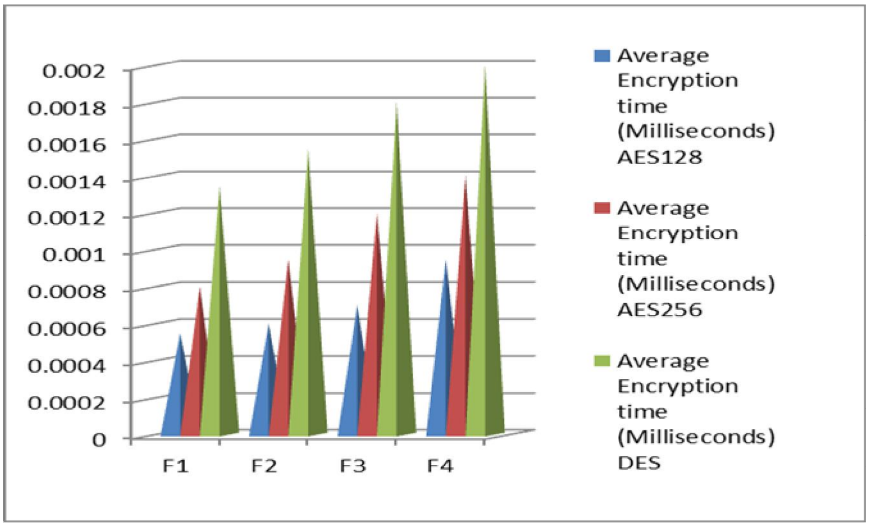


Figure 4: Comparison of three encryption algorithms

Figure (4) shows the result of comparison of different encryption algorithms such as Advanced Encryption Standard (AES128 & AES256) and Data Encryption Standard (DES). The obtained results from different encryption algorithms are used to analyze the result of proposed system. The obtained results from encryption algorithms are compared & after comparing the obtained results it shows the AES128 is efficient algorithm for encryption and decryption.

VII. CONCLUSION

We have implemented the system to provide data security and ensure the correctness of user's data in cloud. Security is provided through the encryption and decryption technique. Also system is implemented for parallel processing of data. We provide dynamic resource allocation by creating virtual nodes and data recovery in cloud computing. So user can recover his/her data if infected or corrupted.

ACKNOWLEDGEMENT

I would like to thanks my guide Prof.A.B.Rajmane for his valuable and constructive comments. I would also like to thanks Prof. Mrs. A.N. Mulla, HOD Computer Science & Engineering Department, Annasaheb Dange College of Engineering & Technology, Ashta for her valuable support.

REFERENCES

[1] Amazon.com. Amazon simple storage service (Amazon S3), 2009.
[2] Cong Wang, Qian Wang, Kui Ren and Wenjing Lou "Ensuring Data Storage Security in Cloud Computing" ,IEEE 2009.
[3] Daniel Warneke, Member, IEEE, and Odej Kao. Dynamic resource allocation for efficient parallel data processing in cloud, 2011.
[4] D. Warneke and O. Kao, "Nephele: Efficient Parallel Data Processing in the Cloud," Proc. Second Workshop Many-Task Computing on Grids and Supercomputers (MTAGS '09), pp. 1-10, 2009.
[5] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. Provable data possession at untrusted stores. In *14th ACM CCS*, pages 598–609, 2007.
[6] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway. UMAC: Fast and secure message authentication. In *CRYPTO*, volume 1666 of *LNCS*, pages 216–233, 1999.
[7] K. D. Bowers, A. Juels, and A Oprea: A high-availability and integrity layer for cloud storage, 2008. IACR ePrint manuscript 2008/489.
[8] K. D. Bowers, A. Juels, and A Oprea. Proofs of retrievability: Theory and implementation, 2008. IACR ePrint manuscript 2008/175.
[9] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L.Youseff, and D.Zagorodnov, "Eucalyptus: A Technical Report on an Elastic Utility Computing Architecture Linking Your Programs to Useful Systems," technical report, Univ. of California, Santa Barbara, 2008.