International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161
©2014 INPRESSCO®, All Rights Reserved
Available at http://inpressco.com/category/ijcet

Research Article

Security and Privacy using Multicloud

P.S.Dhule[Á*] and S.V.Kulkarni[Ḃ]

[Á]Department of Computer Science and Engineering, [Ḃ]Department of Information Technology, DR.BAMU, India

Accepted 10 May 2014, Available online 30 June 2014, **Vol.4, No.3 (June 2014)**

Abstract

In recent years use of Cloud computing in different mode like cloud storage, cloud hosting, cloud servers are increased in industries and other organization as per requirements. While considering the power, stability and the security of cloud one can't ignore different threats to user's data on cloud storage. Data access control is an effective way to ensure the data security in the cloud. However, due to data outsourcing and untrusted cloud servers, the data access control becomes a challenging issue in cloud storage systems. Existing access control schemes are no longer applicable to cloud storage systems, because they either produce multiple encrypted copies of the same data or require a fully trusted cloud server. Malicious user at cloud storage is become most difficult attacks to stop. In proposed system we are implementing the concept of multiple cloud storage along with enhanced security using encryption techniques where rather storing complete file on single cloud system will split the file in different chunks then encrypt and store it on different cloud and the meta data required for decrypting and rearranging a file will be stored in metadata management server.

Keywords:*Cloud,Multicloud, saas,aas,paas,Security , intigirty, confidential,encryption,splitting .*

1. Introduction

Cloud computing offers dynamically scalable resources provisioned as a service over the internet. The third party, on-demand, self-service, pay-per-use, and seamlessly scalable computing resources and services offered by the cloud paradigm promise to reduce capital as well as operational expenditures for hardware and software. It will concentrate on public clouds, because these services demand for the highest security requirements. It also includes high potential for security prospects. It can provide a survey on the achievable security merits by making use of multiple distinct clouds simultaneously. Various distinct architectures are introduced and discussed according to their security and privacy capabilities and prospects.

Jens-matthias bohli, nils gruschka, meiko jensen, member, ieee, luigi lo iacono, and ninja marnau propose security and privacy-enhancing multicloud architectures security challenges are still among the biggest obstacles when considering the adoption of cloud services. This triggered a lot of research activities, resulting in a quantity of proposals targeting the various cloud security threats. Alongside with these security issues, the cloud paradigm comes with a new set of unique features, which open the path toward novel security approaches, techniques, and architectures. This paper provides a survey on the achievable security merits by making use of multiple distinct clouds simultaneously. Various distinct architectures are introduced and discussed according to their security and privacy capabilities and prospects.

The risk for data and applications in a public cloud is the simultaneous usage of multiple clouds. Several approaches employing reducing this paradigm have been proposed recently. They differ in partitioning and distribution patterns, technologies, cryptographic methods,and targeted scenarios as well as security levels. This paper is an extension of and contains a survey on these different security by multicloud adoption approaches. It provides four distinct models in form of abstracted multicloud architectures. These developed multicloud architectures allow to categorize the available schemes and to analyze them according to their security benefits. An assessment of the different methods with regards to legal aspects and compliance implications is given in particular.The idea of making use of multiple clouds has been proposed by bernstein and celesti. However, this previous work did not focus on security. Since then, other approaches considering the security effects have been proposed. These approaches are operating on different cloud service levels, are partly combined with cryptographic methods, and targeting different usage scenarios. In this paper, they introduce a model of different architectural patterns for distributing resources to multiple cloud providers. This model is used to discuss the security benefits and also to classify existing approaches. In our model, we distinguish the following four architectural patterns:

- Replication of applications:- Allows to receive multiple results from one operation performed in distinct clouds and to compare them within the own

*Corresponding author: **P.S.Dhul**

premise. This enables the user to get evidence on the integrity of the result.

- Partition of application system into tiers :-Allows separating the logic from the data. This gives additional protection against data leakage due to flaws in the application logic.
- Partition of application logic into fragments :-Allows distributing the application logic to distinct clouds. This has two benefits. First, no cloud provider learns the complete application logic. Second, no cloud provider learns the overall calculated result of the application. Thus, this leads to data and application confidentiality.
- Partition of application data into fragments:-Allows distributing fine-grained fragments of the data to distinct clouds . None of the involved cloud providers gains access to all the data, which safeguards the data's confidentiality. Each of the introduced architectural patterns provides individual security merits, which map to different application scenarios and their security needs. Obviously, the patterns can be combined resulting in combined security merits, but also in higher deployment and runtime effort.

1.2 Necessity

There are some issues related to single-Cloud model

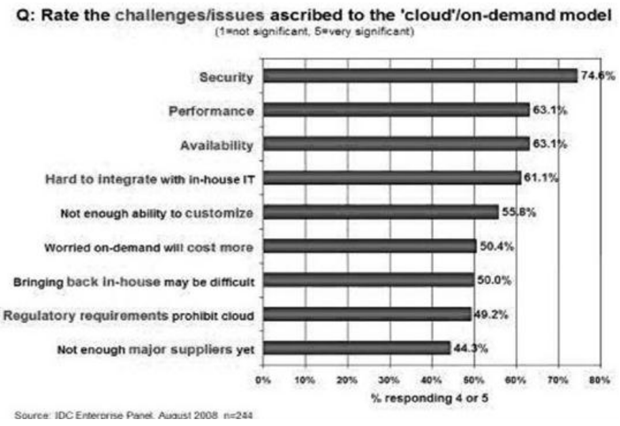
- Malicious system administrator
- Failure of service
- Untrusted cloud provider
- Data integrity loss
- Data intrusion problem

To overcome this issue there is necessary of multicloud model. In our system we are implementing the concept of multiple cloud storage along with enhanced security using encryption and splitting techniques.

1.3Objectives

Till 2010, eighty percent research was carried on single clouds whereas only 20 percent was done in multi-clouds. In Multi-Cloud environment services are improved by distributing reliability, trust and security among various cloud providers **.**Adversary never get complete set of data in multi-cloud environment thus removes most of threats in single cloud model Distributed File System (DFS) is used in multi-cloud model to divide data into multiple nodes for parallel execution of Map Reduce operations. Our objective is to design and implement multi-cloud based secured storage system using DFS mechanism .It will be hosted on exiting public cloud infrastructure. It will divide cloud user's data into multiple chunks, which will be deployed in multiple clouds in secured manner.

From one point of view, security could improve due to centralization of data and increased security-focused resources. On the other hand concerns persist about loss of control over certain sensitive data, and the lack of security for stored kernels entrusted to cloud providers. If those providers have not done good jobs securing their own environments, the consumers could be in trouble. Measuring the quality of cloud providers' approach to security is difficult because many cloud providers will not expose their infrastructure to customers. This work is a survey more specific to the different security issues and the associated challenges that has emanated in the cloud computing system.



Source: IDC Enterprise Panel, August 2008 n=244

Figure 1: Results of IDC survey ranking security challenge.

2. Literature Review and Related Work

Several researchers studied security challenges and proposed various defense mechanisms related to Cloud computing models. In this section, we conduct a brief study of relevant work done. National institute of standards and technology, USA and Cloud security alliance has given basic information about various security challenges in cloud computing paradigm and also suggested mechanism to be followed for better security .

Zhifeng Xiao and Yang Xiao described in their survey paper various security vulnerabilities, threat models and respective defense mechanism for confidentiality, availability, integrity, privacy-preservability and accountability. They have proposed various threats like Cross-VM attack, malicious system administrator for cloud confidentiality which can be defended by VM placement prevention, No hypervisor model and use of trusted computing. Cloud Integrity is affected by data loss or data manipulation and dishonest computation in remote cloud provider server which can be managed by third party auditing, replication, recomputation. Cloud Accountability is affected by SLA violation and dishonest Map Reduce which can be deal with accountable SLA and accountable Map Reduce. Even though authors had given complete detailed survey of cloud threats and vulnerabilities but they have not mentioned which method will be suitable for multi cloud environment.

Single cloud environment is affected with service unavailability, malicious system administrator and data integrity challenges which can be solved by the use of multiple clouds. Mohammed A. Alzain et al. in [4] described multi-cloud model as combination of various cloud where user data will be distributed and executed in those clouds simultaneously. It is observed that multi–clouds improve performance provided by single cloud environment by dividing security, trust and reliability among different clouds. They have made a survey of various techniques available for multi cloud security like use of cryptography, secret sharing algorithm, DepSky system, redundant array of cloud storage (RACS) and HAIL protocol. They have shown limitations of exiting solutions and suggested for secure cloud database as their future work but they had not given their solution in details.

Mukhesh Singhal et al. proposed multi-cloud computing framework using proxy VM instance for sharing resources and dynamic collaboration among cloud based services. This framework manages security, mutual trust an policy issues without need of pre-collaboration agreement, which is necessity in cloud mashups. Whenever cloud user wishes to use any services, he will send request to cloud where CSP has pre-installed proxy VM which will interact with multi-cloud services and provide results to user. It helps for collaboration among various cloud user .They have provided different proxy architecture as cloud hosted proxy, proxy as a service and on premise proxy of which first two architecture are cloud service provider and proxy service provider dependent which cannot resolves malicious system administrator problem. Peer -to Peer proxy architecture will be more secure where client have control over proxies.

Bohli J. et al. have proposed 4 different architecture for multi-cloud computing paradigm for improving security and privacy of user and provider. First approach specifies replication of application which helps to verify integrity of data after execution in cloud is over. Second approach helps to protect data and logic by separating them in third approach data and application confidentiality by breaking application logic in parts and executing it over multiple clouds. Similar approach is given in last architecture where data is broken into parts and executed over various clouds which helps to protect from malicious cloud service provider. Each archit1ecture has their own pros and cons however combination of that architecture will give better secure approach for multi-cloud systems.

After this reviews, we came to conclusion that we can attain more security by distributing user data and application among multiple clouds with full control to user at SaaS level where comes our proposed approach based on distributed file system. A distributed file system is used to support storing and sharing of files of multiple users physically distributed in distributed network with location transparency. A DFS can solve purpose of secure storage with reducing most of challenges faced by single cloud.

In authors had given survey of various popular DFS like Google GFS, NFS, Andrew, KFS, Hadoop etc with their advantages and disadvantages based on features of architecture, processes, replication, fault tolerance and security. Survey shows that none of available DFS are fully secure.

Paval Bzoch and Jiri Safarik in studied traditional and modern trends in security and reliability features of DFS. Journal technique is used for reliability by storing all changes to file system in journal and flush it with synchronization once changes are made permanent. It can be used for checkpoint purpose for fault recovery. Replication methods like master-slave, 3-way replication are discussed. Authors suggested that security of DFS can be achieved by secured network connection and secured file storage by using various cryptographic techniques.

In Dalibor Peric et al. Proposed DRFS, distributed reliable P2P file system with dynamic active replication and used random and independent identifier for data storage but this system will not be suitable for cloud dynamic environment.

In decentralized Access control mechanism to improve scalability and high availability of distributed file systems. They have used authorization certificate and decentralized certificate revocation list with consistent hashing. In distributed system nodes data is distributed in various nodes remotely. Nodes may be upgraded or may get failed. Files can be added/deleted in run time dynamically.

This creates load imbalance in DFS. In new distributed loads rebalance approach is presented. Every chunk server node without having global knowledge checks whether it is overloaded or under loaded by comparing load against threshold value, then overloaded node can share its tasks with nearest lihtweighted node and lightest node leaves system by transferring its tasks to its successor node. This approach is better than traditional centralized load balancing node approach in exiting DFS like Google GFS and Hadoop HDFS. Sandesh Uppoor et al. introduced cloud based synchronization of distributed file system hierarchies. The authors have suggested that it is based on P2P synchronization of data which requires less cost and bandwidth as compared with cloud based master-replica synchronization approach. Even though distributed file systems are not new and much research was carried out till now but it is based on improving features of DFS like security, reliability and scalability .Very few research was carried out on use of DFS as secure storage system in cloud environment. In revised Blakely's secret sharing mechanism is proposed to improve security and reliability of DFS without affecting scalability. This scheme does not require key management .To reduce computation overhead in this scheme, Graphical processing unit is used.

Even though cryptographic key management is required in this approach but cost of implementation is increased due to GPU also authors have tested this scheme in simple environment with 8 CPU core and actual testing is not done in cloud. Authors have not mentioned how DFS will communicate with other proprietary DFS in multi cloud environment.

Fan-Hsun et al. proposed secure and reliable cloud DFS using replacement of Hadoop DFS with open source based Tahoe least-authority file system. This system improves fault tolerance by recovering data even though some storage nodes are faulty. It is more secure as Tahoe-LAFS incorporates AES encryption.

This system is secure and reliable but authors have used already implemented Tahoe-LAFS and new DFS is not developed. This DFS is tested only on 4 storage nodes and no discussion was carried out by authors about use of system in multi-cloud environment and operations supported by File system.

In cloud storage service model for inter and intra cloud is proposed at Iaas level. Different data chunks of file are stored in various VMs of single or multi cloud. User can store and retrieve data from multiple cloud .System uses user authentication followed by file splitting / file retrieval by cloud manager interface and then it will be handover to multiple clouds. System supports both inter and intra cloud operations but security is not enough as encryption methodology is not used. Authors have not discussed different threats at IaaS level to virtual machine like cross-VM attack and side channel attack.

Kheng Kok Mar in introduced multiple cloud based secure virtual diffused file system by hosting it on exiting setup of public cloud. This system used information dispersal algorithm to divide data into multiple parts and diffused them in various clouds. It used registry server for managing metadata and data distribution. This DFS scheme supports random read/ write and streaming I/O operations.

Even though system is good for operating at multi-cloud environment, it is using only data splitting architecture, one of four architecture proposed in. System is less secure as compulsory encryption is not used.

3. Proposed System Architecture

Figure 2 shows the detailed architecture of the proposed system, which describe the detailed process flow. Using the Dot net technology we will first build a data accessing classes for different cloud storage using cloud specific proprietary access scheme. These classes are not directly exposed to the end user or developer. After this we will develop a wrapper classes which are exposed to the developer in order to write his software program to access data to and from cloud storage. First level of development will be platform as a service (PAAS) and second level of development will be Software as a service (SAAS)

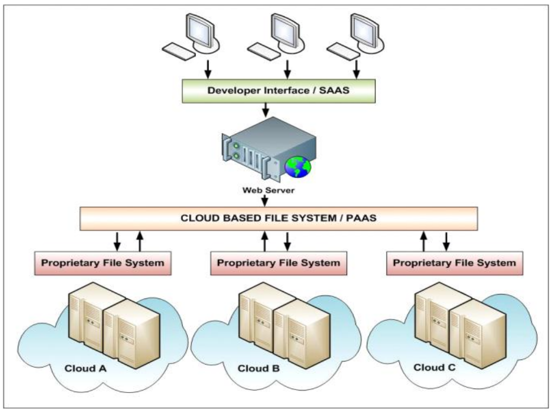


Figure 2: Proposed System Architecture

- Setting up and configuring different cloud server in order to having storage cloud access
- Using cloud server API develop file accessing method in different cloud.
- Developing encryption techniques like AES, RSA for file decryption before storing it on cloud.
- Develop a file management classes in dot net.
- Develop a web interface to upload and download files in cloud storage.

4. Methodology

4.1 Registration

In registration get username, email address , password, user generate random verification code (say cd)are as NewRandom.next(0/9).

NewRandom.Next():-function can generate character its six time generate and concatenate the character then execute .we get 6 digit random code.Check existing email address by executing select query where email address is equal to provided value by user otherwise get zero).

First login verification code ask status =deactive

insert into table all fields with status as deactive and confirmation code=cd.

Send mail to user email address by using SMTP mail class from .NET use parameter for mail:-sender,receiver ,username ,pass message , subject,message body. Create folder with userID as folder name using directory.createdirectory(path) function.(act as personal folder to user) and first part of file will be stored here.

4.2. Login

In this form get userID and password from user execute.

*from table where userID=”given by user” and password=”given by user”

If this query execute return data , it means user name and password is valid.For this we use SQLconnection.class to establish connection with database.Next we use command to execute command given sql query . This command will return which will connect SqlDataReader.

SqlDataReader.Read: Method is used to read the records.

Get user status from this query

If(DataReader(“status”)=”deactive”)

Then redirect to verification code else redirect to user home.While redirecting to user home, store current userID in session variable. This will allow application to know which user is logged in.

4.3. Verification

Now in this step get the logged username from session variable. If session variable is blank then redirect to login page else execute sql query and confirmation of logged in user from table(a). get confirmation code in textbox(b)

Now compare a with b, if matched then update user status field with text active and redirect user to the home page

4.4. Setting page

As our file get distributed at three different location we have one location that is our application. Now we need two more FTP where 2nd and 3rd file we stored, so we design setting page where this will be further used by application to upload and download file from created table.Insert into table FTP details.

4.5. Homepage

Homepage will show list of file uploaded. By user from user specific directory . here we use from user specific directory,here we use data list to show file listing and system.directory and system.file class to get folder and file details like file name , file size.

- Upload file by using file uploader control we can let the user select file to be upload.
- Get the sever path by using ServerMappath function to get path of server directory.

4.6. Encryption

Open file in memory string using System.IO.MemoryStream class, This will convert file into memory stream of bytes then encrypt the file used. System.Cryptography class

From .net tripleDES technique we need to pass 24 byte encryption key to this method. This key will provided by textbox(UI). Now encrypt the file and collect in another file.

Note: we are getting public key from user itself.

4.7. Splitting

.Net provide a function

<u>1stfunction</u>: System.MemoryStream.2array.Take(length) To get this 1st byte of given length from any memory stram.

<u>2ndfunction:-</u>

System.MemoryStream.2array.Skipp(length)

This function will skip 1st given length byte and return memory stream.

File.Write.AllBytes(filename,memorystream)

This will save memory stream as file to disk. Save file1in user directory using File.Save(file a) function. Now get 1st FTP password from user in text code connect to 1st FTP server using FTP.connect function.Get FTP URL and username from table. Upload file to user directory using FTP.save function i.e. second part of (B)FTP.Disconect Same process do for 2nd FTP process.

4.8. Download

Get the file name selected by user read 1st part of file(means file a) using FileStream function from user specific directory. You will get A now get FTP detail from user get from user name and FTP password user in textbox connect B FTP download 2nd part from FTP using FTPWebResponse class or method or we can use WebRequestMethod.FTP. Download file function now we get part B and repeat above process we will get C or part C. now using Buffered.Copy function combine 2nd (B) and 3rd (C) part we will get X, then combine i.e. 1st part with X . Finally we have club file in Byte buffer . now save this buffer to memory Stream.

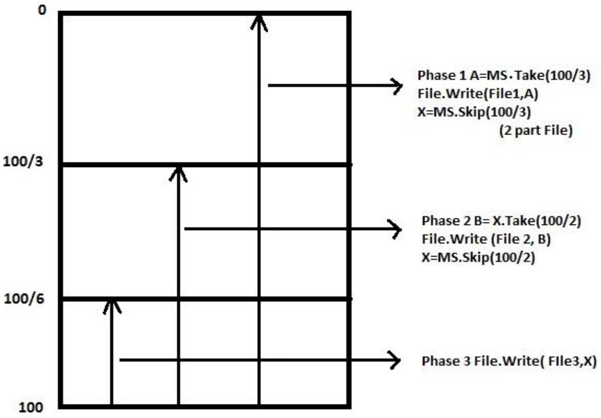


Figure 3: Splitting And Merge

4.9.Decrypt

Get the public key i.e. encryption key from textbox and using TripleDES decryptor class decrypt the memory stream. Now save this memory stream using File.Write.Allbyte function to sever disk in temporary function and redirect web client i.e. browser to this Temp file and browser start download file.

5. Work Plan

Phase I

Here in first phase all the existing approaches in same system stream a different applications and the outcome of the currently working file system with their advantages and disadvantage get studied

Phase II

The proposed idea is to design and implement cloud based encrypted file system and test it over real time application so here in this phase we will further research and test existing file system and try to develop different sample software program by referring existing open source software. In this phase we will also test the feasibility of sample programs on different platform like PC, Laptop, Mobile devices, etc. all the collected results and analysis will be collected in this phase.

Phase III

Once we are ready with the results and analysis now we will design and software process and architecture based first version of software model which will meet the defined goal of proposed system. This phase will deal with core functionality of the system and once this get ready and tested then we can go with the security aspects of the system.

Phase IV

After the testing core functionality of the proposed system here in this phase we will add a security approach and enhance the system by testing it in real environment. Here actual execution and the feasibility of the system will be tested over private and the public cloud using custom designed software. Once this is done we can publish released developer version of the system.

Phase V

As we are developing and development platform cum application programming interface we need to design and develop different sample code or application to test the stability of core function with development environment.

Conclusion

Once the system is ready it is expected that it should give the desired output of easy and efficient data access by the way of splitting and securing the data at multiple cloud environments without the knowledge of the end user. It should also expect that system should be capable to restore the data from remaining cloud storage in case of failure of any of the cloud storage. Another import ant expectation from the system is that it should be developer's friendly so that vast use of the platform by number of developers and it will results into multiple implementation of the system.

References

Lee Badger, Tim Grance, Robert Patt-Corner, Jeff Voas (May 2011) DRAFT Cloud Computing Synopsis and Recommendations, *NIST Special Publication 800-146,*

Cloud Security Alliance (CSA) Released December 17, 2009). Security Guidance for Critical Areas of Focus in Cloud Computing V2.1,http:// www.cloudsecurityalliance.org/guidance/ csaguide.v2.1.pdf

Zhifeng Xiao and Yang Xiao(March 2012), Security and Privacy in Cloud Computing,*IEEE Communications Surveys & Tutorials*

Mohammed A. AlZain, Eric Pardede, Ben Soh, James A. Thom(2012), Cloud Computing Security: From Single to Multi-Clouds,*IEEE 45th Hawaii International Conference on System Sciences*

Singhal M., Chandrasekhar S., Tingjian Ge., Sandhu R., Krishnan R., Gail-Joon Ahn., Bertino E(Feb 2013), Collaboration in Multicloud Computing Environments: Framework and Security Issues, *IEEE computer society journal, Vol. 46, Issue 2, pp. 76-84*

Bohli J., Gruschka N., Jensen M., Lo Iacono L., Marnau N(2013),Security and Privacy Enhancing Multi-Cloud Architectures,*IEEE Transaction on Dependable and secure computing, Vol PP, Issue 99*

Tran Doan Thanh, Subaji Mohan, Eunmi Choil, SangBum Kim, Pilsung Kim (2008)A Taxonomy and Survey on Distributed File Systems, *IEEE Fourth International Conference on Networked Computing and Advanced Information Management*

Paval Bzoch, Jiri Safarik(Sep 2011)Security and reliability of distributed file systems, *6th IEEE international con. on intelligent data acquisition and advanced computing systems*

Dalibor Peric, Thomas Bocek, Fabio Victora Hecht, David Hausheer, Burkhard Stiller(2009), The design and evaluation of a distributed reliable file system, *Int. Conference of parallel and distributed computing, application and technologies*

Jumpei Arakawa, Koichi Sasada(December 2011),A decentralised access control mechanism using authorization certificate for distributed file systems, *6th Int. Conference on internet technology and secured transactions, UAE*

Hung-Chang Haiao, Hsueh –Yi Chung, Haiying Shen, Yu-Chang Chao(May 2013),Load rebalancing for distributed file systems in clouds, *IEEE transactions on parallel and distributed systems, Vol. 24, No. 5*

Hadoop Distributed File System, *http://hadoop.apache.org/hdfs/2012*

Satyanarayanan, M".(1989), A Survey of Distributed File Systems,*Technical Report CMU-CS-89- 116, Department of Computer Science, Carnegie Mellon University*

Sandesh Uppoor, Michail D. Flouris, Angelos Bilas(2010),Cloud-based synchronization of distributed file system hierarchies, *IEEE*

Paval Bzoch(June 2012), Distributed File Systems, *Technical Report no. DCSE/TR-2012-02, University of West Bohemia,*

Su Chen, Yi Chen, Hai Jiang, Laurence T Yang, Kuan-Ching Li(2012), A secure distributed file system based on revised Blakely's secret sharing scheme, *11th IEEE international conference on trust, security and privacy in computing and communications*

Fan-Hsun Tseng, Chi-Yuan Chen, Li-Der Chou, Han-Chieh Chao(November 2012),Implement a reliable and secure cloud distributed file system*, IEEE international symposium on intelligent signal processing and communication systems*

Shushant Shrivastava, Vikas Gupta, Rajesh Yadav, Krishna Kant(2012), Enhanced Distributed storage on the cloud,*IEEE 3rd international conference on computer and Communication technology*

Kheng Kok Mar(December 2011),Secured virtual diffused file system for the cloud,*6th International IEEE conference on internet technology and secured transactions, UAE*