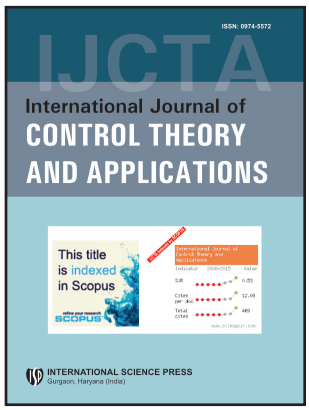

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 24 • 2017

Enhanced Framework for File Transfer Service to Ensure Confidentiality in Public Cloud Environment

Athira G. Panicker[a], Reshma V.R.[b] and Nima S. Nair[c]

[a,b]PG Student, Department of Computer Science & IT, Amrita School of Arts and Sciences, Kochi, Amrita Vishwa Vidyapeetham (Amrita University), India. Email: [a]athiragpanicker93@gmail.com, [b]reshmavraghav26@gmail.com

[c]Asst. Professor, Department of Computer Science & IT, Amrita School of Arts and Sciences, Kochi, Amrita Vishwa Vidyapeetham (Amrita University), India. Email: nimasnair@gmail.com

Abstract: Cloud computing is a new era of computing where storage and computing resources can be used on a metered basis. These resources are made available to users as on- demand services and thus reduces the investments in an organization's computing infrastructure. Because of this splendid feature of cloud computing, nowadays a lot of users store their countless number of confidential files in the cloud storage server. So that it is the responsibility of the storage system to provide powerful confidentiality measures to those files. In this paper, we have proposed a security enhancement framework model which gives more confidentiality to files. To achieve this we introduce splitting and shuffling operations. These operations are implemented in cloud storage server.

Keywords: Cloud computing; File transfer; confidentiality; Splitting; Shuffling.

1. INTRODUCTION

Cloud computing permits business consortiums and individuals to enjoy on-demand service, ubiquitous network access, resource pooling, rapid elasticity and measured services anywhere at any time. One of the prominent services offered by cloud computing is the cloud data storage in which the user can store their valuable data on cloud service provider's server instead of their own servers. The users only need to pay for the service that they consume. Scalability, flexibility, reliability, cost efficient, on demand self – service etc [1, 2] are the features of cloud computing. The user does not need the entire software and hardware infrastructure for their use. All those will be provided by cloud service providers. The three different services offered by cloud computing are Software as a Service (SaaS): it means a complete application running on someone else's system, Platform as a Service (PaaS): it helps the applications using web based tools so they run on systems software and hardware provided by another company and the third one is Infrastructure as a Service (IaaS): provide access to raw computing hardware over the internet; such as servers or storage [3].

Today cloud computing increases its popularity as more individuals uses services like Google drive for data storage, Organisations and companies outsource their data to Amazon web service etc. Data owners trust the service providers on keeping their valuable data on cloud. So it is important for the service provider to keep those data secure [4]. Data protection is one of the core problem in cloud computing. It deals with 4 major security aspects: availability, confidentiality, integrity and authentication. Data confidentiality ensures that only authorized person is using the data. Data integrity refers to information that has not been modified or remains untouched. Authentication refers to the process of verifying whether the incoming user is authorized or not. Data availability refers to the ability to guarantee the use of data in time when needed and also refers to the availability of Cloud service provider on-demand [1].

In this paper we focus on preserving data confidentiality. Confidentiality can be used to protect the data from unauthorized access from both inside and outside attackers [3]. Since the service providers store different user's data, access to those data should only provided to authenticate one. If the confidentiality parameter is broken then the organization will lose their data. So it is the duty of service providers to maintain this security parameter throughout the file transferring process. Confidentiality can be achieved through cryptographic encryption techniques. Encryption is the practice of transforming comprehensible text into unreadable form using an algorithm and a key [6, 7]. Usually, confidentiality is maintained by encryption techniques. Encryption alone cannot provide data protection in cloud environment. Various techniques exist for maintaining confidentiality such as; the data to be stored in the cloud is divided into different fragments and distributed to different service providers. Another method is applying both encryption and obfuscation on data. Our model preserves confidentiality of files by encryption, splitting and shuffling [8] operations.

2. RELATED WORK

Ensuring data confidentiality in cloud data storage is one of the challenging issues. Many approaches and security protocols have been suggested for providing storage security in cloud environment. Most of the papers suggest encryption as a good technique.

A security model suggested for maintaining confidentiality uses a combination of two techniques: encryption and Obfuscation [5]. Encryption is applied to alphabets, alphanumeric and symbols. Obfuscation technique is applied to numeric type of data. They suggest that applying encryption and obfuscation techniques on the cloud data will provide more protection against unauthorized usage. That is confidentiality could be achieved with a combination of encryption and obfuscation. All the data is either encrypted or obfuscated before it is sent to the cloud database. Encryption and obfuscation of cloud data is done at the user side. For this they use five different algorithms. Algorithm 1 for finding type of data whether it is numeric or alphabets. Second algorithm is for obfuscation and the third one is for encryption. Algorithm 4 generates a random integer and the last one for generating random string value. In this model the data to be stored in the cloud should be encrypted or obfuscated before it sends to the cloud storage. Both the encryption and obfuscation are done at the client side itself.

Another approach to provide security and confidentiality in cloud storage environment is through fragmentation techniques [9]. Privacy and data availability can be ensured by dividing the user's data block into data pieces and distributing them among the available service providers. This model provide the cloud computing users a decision model, that provides a better security by distributing the data over multiple cloud service providers in such a way that, none of the service provider can successfully retrieve meaningful information from the data pieces allocated at their servers. This model provides user with better availability of data.

The information decomposition and dispersion of separate data into unrecognizable parts and store them in distributed hosts in the cloud also preserves confidentiality [10]. Data colouring method based on cloud

watermarking solves the trust management issue between data owners and storage service providers [11]. The watermarking technique is suggested to protect shared data objects and massively distributed software modules. These techniques safeguard user authentication and tighten the data access-control in public clouds.

3. PROPOSED SYSTEM

The aim of our proposed work is to enhance the security measures of file transfer in cloud computing. We mainly focus on the confidentiality of file. During file transferring, the possibility of unauthorized access is high. This will compromises the trust level between file owners and cloud service providers. So it is important to make sure that during file transfer, the file can be accessed only through right hands. The overall framework is depicted in Figure 1.

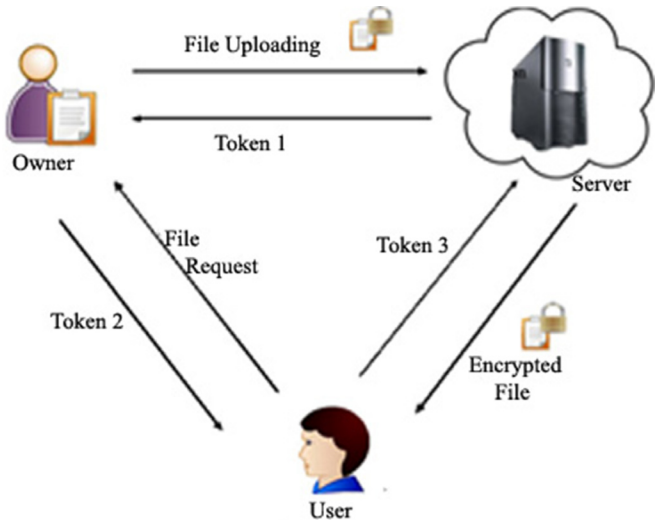

Figure 1: Security Framework

We experimented with three nodes; the server node, owner node and the user node. The server node is the place where the file is stored. The owner node is one who owns the file. The user node is who gains file access permission from owner. As we mentioned earlier there are two operations performed in our framework. The owner uploads his file using his username and password. Only the authenticated owner can upload his file into the cloud. Before the owner uploads his file to the server, to ensure more security, the owner encrypts the file by using RSA algorithm. When the file is successfully uploaded to the server it will produce a token. The token consists of owner id, file name and the corresponding file id in the encrypted form by using owner's public key. This token (Token 1) will be forwarded to the owner. The owner decrypts it using owner's private key. The owner keeps this decrypted token and list of all the privileged users. If a user wants a file he/she will first sends a request to the owner. When the owner receives a file request from the client; the owner checks whether the user has privileged permission to access the requested file. If it is, the owner creates a new token (Token 2) and a key (key is a secret key between owner and the user) and sends it to the user, the token consists of owner id, file name and file id. This token will be encrypted using a shared secret key between owner and the server. After this, token and the key will be encrypted (termed as envelop) using owner's private key and then forwarded to the corresponding user.

The user decrypts this envelop using owner's public key. The key will be kept by the user aside itself and the token (Token 3) will be send to the server. After getting the token (Token 3), the server fetch the file based

on the file id in the token. After decrypting the token using shared secret key between owner and the server, the server checks the token values with the file which is stored in the server. If it is matches the server performs two processes before it sends the file to the user. The overall dialogues between user, owner and the server in the system is described in Figure 2.

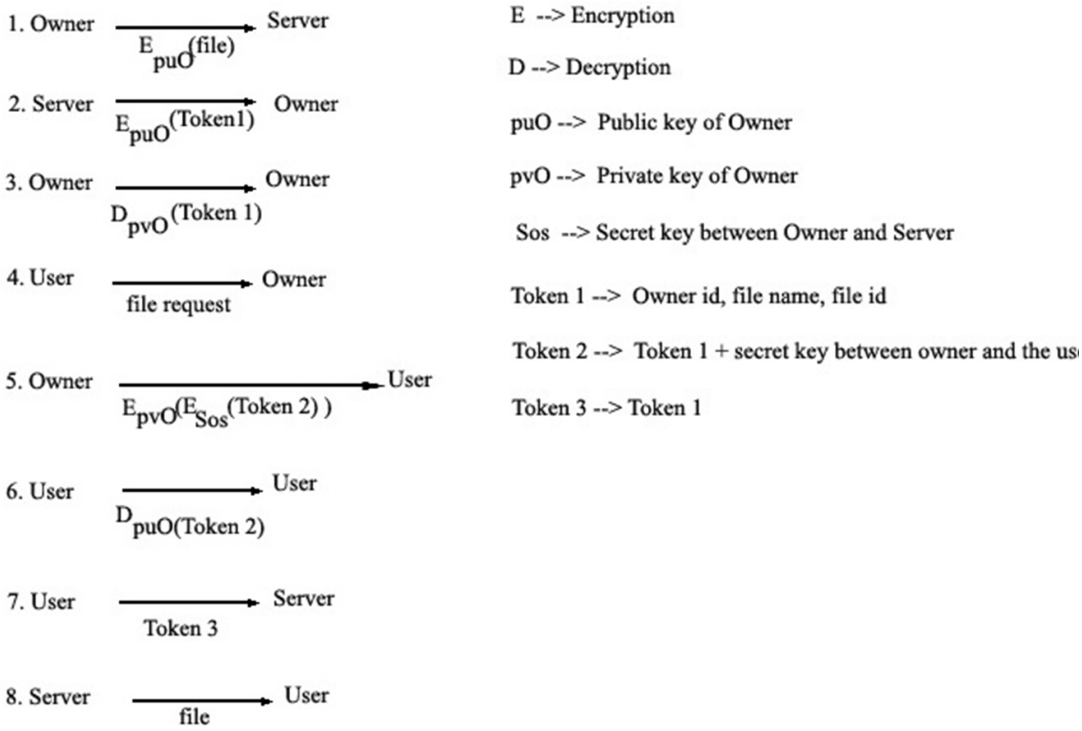

Figure 2: System Dialogues

3.1. Splitting

In the first stage, the server fetches the requested file. After the server splits the fetched file into equal number of blocks which are here termed as buckets. The splitting is done on the basis of file size. All these buckets are filled by same amount of data. There is a chance for last bucket being incomplete. To overcome this situation we add some padding bits to the last bucket. This process is shown by the following algorithm.

1. Read the requested file.
2. Variable length = Length of the requested File.
3. Variable $a = \lceil\sqrt{\text{length}}\rceil$
4. Create $\lceil \text{length}/a \rceil$ buckets of length 'a' each.
5. Split the file into 'a' buckets [a-(length % a)bits at the end of ath bucket add dummy bits].

3.2. Shuffling

After the splitting operation the server performs shuffling operation on different buckets. The algorithm generates a random permutation which is used for shuffling. The order for de-shuffling is attached to the header part of first bucket. The requested user uses this information for de-shuffling. After the shuffling operation the server sends the shuffled blocks to requested user. This process is shown in the following algorithm:

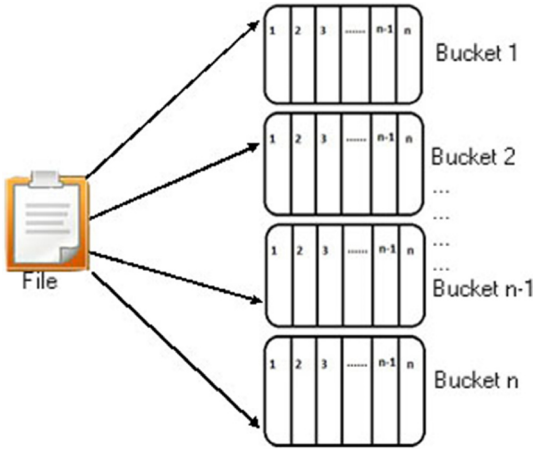

Figure 3: Splitting

1 .Fetch the '*a*' buckets and load these to an array.

2. Variable *n* = Length of the array.

3. Variable random

4. for each element of the array indexed by *i*

    random = a random number in the range 0 to *n* – 1.

    swap the '*i*' element in the array with 'random'.

    end for.

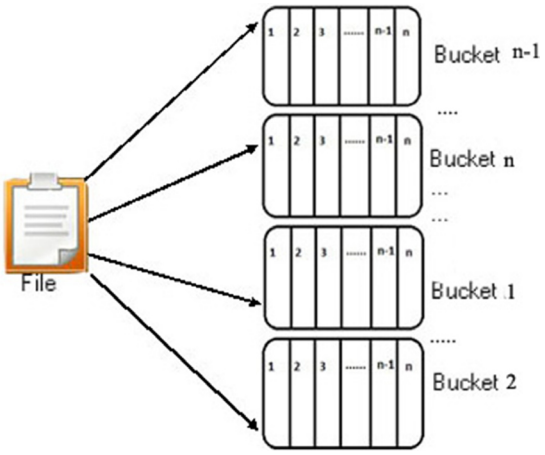

Figure 4: Shuffling Buckets

The user on receiving this data will first reshuffle by using the data defined in header part. After this the user join the different buckets (blocks of data) to form a single file. Then user decrypts the file using the key given by the file owner. Thus user gets the original file that he/she requested. Only the authenticated user can read or receive the file. Thus the security parameter data confidentiality is maintained.

Here we use asymmetric key encryption. RSA is one of the most secured asymmetric key encryption techniques. So we choose RSA for encrypting files and tokens. In cryptography, the key length is directly proportional to encryption strength. RSA has large key length, so it is difficult to break by the attacker. In addition

to RSA algorithm we also use the concepts of splitting and shuffling. For shuffling we used Fisher Yates shuffle algorithm which will give unbiased random permutation. Thus RSA combined with shuffler will increase the confidentiality of the system.

We experimented our proposed work using VMware workstation with minimum three nodes - A server, owner and host. In the server node we installed MS Windows 2008 Server and configured Internet Information Service and FTP service. Our application act as a middleware between client request and server service. Whenever the server starts, our program module for splitting and shuffling automatically gets started and always there in the memory. It listens for a request from the client. Whenever the client requests for a file, our program module will first check whether the user has the privilege to access this particular file or not. If he has, the server fetches the file and splits it into finite number of blocks and changes the memory locations of these blocks in unbiased random order. These operations increase the confidentiality level of our system. The implementation model is described in Figure 5.

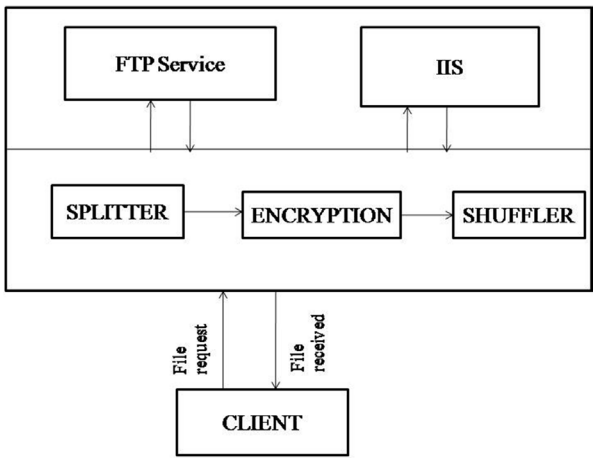

Figure 5: Implementation model

4. CONCLUSION AND FUTURE SCOPE

Today cloud computing become a most popular computing services in IT. Most of the organizations out-source their confidential data to different service providers. So the organization fully trust service provider. Therefore it is the duty of service providers to maintain the confidentiality of data. Our proposed model keep the data secure by encryption, splitting and then shuffling. Only the authenticated user can decrypt, reshuffle and join the file thus preventing unauthorized access to the file. This module works well in client server scenario. Further we will try to implement this model in cloud based environment.

REFERENCES

[1] K. S. Wagh, Rasika Jathar, Sonal Bangar, Anu Bhakthadas: Securing Data Transfer in Cloud Environment Anu Bhakthadas et. al., Int. Journal of Engineering Research and Applications ISSN : 2248-9622, Vol. 4, Issue 5(Version 2), May 2014, pp.189-193

[2] Anurag Porwal, Rohit Maheshwari, B.L.Pal, Gaurav Kakhan: An Approach for Secure Data Transmission in Private Cloud International Journal of Soft Computing and Engineering (IJSCE)

[3] Akansha Upadhyay, Manu Shrivastava: Data Storage Security in Cloud Computing International Conference on Cloud, Big Data and Trust 2013, Nov 13-15, RGPV.1

[4] L. Arockiam and S. Monikandan: Security Framework to Ensure the Confidentiality of Outsourced Data in Public Cloud Storage International Journal of Current Engineering and Technology E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

[5] L. Arockiam, S. Monikandan, P. D. Sheba K Malarchelvi: Obfuscrypt: A Novel Confidentiality Technique for Cloud Storage International Journal of Computer Applications (0975 – 8887) Volume 88 – No.1, February 2014

[6] Tejas P Bhatt: Security in Cloud Computing Using File Encryption IJERT ISSN: 2278-0181 Vol. 1 Issue 9, Nov-2012

[7] Kawser Wazed Nafi, Tonny Shekha Kar, Sayed Anisul Hoque, Dr.M.M.A Hashem: A Newer User Authentication, File encryption and Distributed Server Based Cloud Computing Security Architecture (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 10, 2012

[8] Olga Ohrimenko, Michael T. Goodrich, Roberto Tamassia, Eli Upfal: The Melbourne Shuffle: Improving Oblivious Storage in the Cloud arXiv: 1462.5524V1 [cs.CR] 22 Feb 2014

[9] Thota Reshma Kishore, D.Akhila Devi, S.Prathyusha, D. Bhagyasri, Bhuma Naresh: Client and Data Confidentiality in Cloud Computing Using Fragmentation Method International Journal of Soft Computing Engineering (IJSCE) ISSN: 2231-2307, VOLUME-3, Issue-2, May 2013

[10] Eliseu Castelo Branco, Javam de Castro Machado, Jose Maria da Silva Monterio Filho: A Strategy to Preserve Data Confidentiality in Cloud Storage Services 29th SBBD-WTDBD-ISSN 2316-5170

[11] Nagaram Ramesh, B.Nagaveni, P.Satyavathi: An Efficient Technique to Provide Security for Data Owners in Cloud Computing International Journal of Research & Technology (IJERT) ISSN: 2278-0181 Vol.1 Issue 5, July-2012