Scientific Journal of Impact Factor (SJIF): 4.14
e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 3, Issue 5, May -2016

Cryptography based key reduction and security for file access on cloud.

Prof. Pavan Kulkarni[1], Shreyas Kavathekar[2], Gaurav Barfe [3], Mahesh Dhokade[4], Faisal Khan[5]

[1] Asst. Prof., Department of Computer Engineering, TCOER, Pune University,Pune, Maharashtra, India
[2] Student, Department of Computer Engineering, TCOER, Pune University,Pune, Maharashtra, India
[3] Student, Department of Computer Engineering, TCOER, Pune University,Pune, Maharashtra, India
[4] Student, Department of Computer Engineering, TCOER, Pune University,Pune, Maharashtra, India
[5] Student, Department of Computer Engineering, TCOER, Pune University,Pune, Maharashtra, India

ABSTRACT- *Cloud industry is growing with tremendous pace. It is the result of its various features and services being offered. Along with big companies even the minnows and start-ups are willing to take advantage of these facilities. But as usual every technological revolution brings some challenges and it becomes necessary to find solutions to them. One of the biggest challenges in this particular technology is security, security of data! There are many attempts made to provide solution to this challenge; some of which have used cryptography in various ways. Cryptography has proved to be a good safety measure but it brings several challenges with it. One of which is the large number of cryptographic keys. Attempts are being made to reduce the number of keys and we aim for the same.*
A convention is proposed in this paper which helps to reduce the number of keys without compromising the security by using cryptographic password breaking method.

Index Terms- *Cloud computing, cryptography, SHA, security, symmetric keys, Insider threat, Team leader, cybercrime.*

I. INTRODUCTION

The number of companies providing cloud services is increasing day by day. The reason behind this increasing graph is the growing number of users. These days industries always are in search of opportunities to save money and in one of the numerous advantages of migrating to cloud, ‘saving money’ tops the list. But with the number of users increasing, it also attracts cyber criminals. The amount of data available on cloud severs has made it a tempting target for cyber criminals. Big cloud service providers such as Google and Amazon can at time survive a cyberattack, but small cloud service providers may not be able to handle such attacks. [1] And hence cloud security has become such a vital issue in this industry. In this paper we are concentrating more on reducing the overhead which is the result of cryptographic keys without compromising on data integrity. We are using a team leader based password breaking method in which we are not just breaking the password but we are also creating a hash code which will be generated from the entered password and more over the hash is also encrypted. Along with the password’s hash the file is also encrypted using that same hash to ensure protection. The password’s hash will then be decrypted. To decrypt the hash a certain ratio or minimum number of correct passwords must be entered [2]. But the most important part of that code is with team leader as until his correct part of password is not entered the file cannot be decrypted.

The number of keys are reduced in this scheme and most importantly data integrity is maintained as we can compare the hash codes that we are going to use.

II. RELATED PREVIOUS WORK

John, Lori, and Bruce in their paper have perfectly put forward why security is so crucial for cloud service providers. And that resulted in the motivation to work in this area. Their extensive research on the growth of cloud gives us the hint why cloud has become a soft target for cyber criminals. [1]
In their paper Sushil, Sanjeev, Arvendra, Sundaram have proposed a similar scheme and in fact this paper is the basis of our work. We are trying to work on the security part more as the challenge of reducing the number of keys is well handled in the scheme proposed by them. We are putting another anchor character a ‘Team leader’ as the malicious insider can always be a group of certain people. Here we assume that a team leader is well chosen and is loyal to the employer. We will be using a very much similar terminology to this paper as the work is also somewhat similar but sill different on many levels. Our paper is an attempt to advance the version of this paper [2]

Amanjot and Manisha have proposed an approach that uses RSA, Triple DES and Random Number generator algorithms, their approach does provide security but at the cost of heavy overhead in cases of large amount of data. We attempt to overcome this issue by using a single key but by breaking it for a group of large number of users [3]

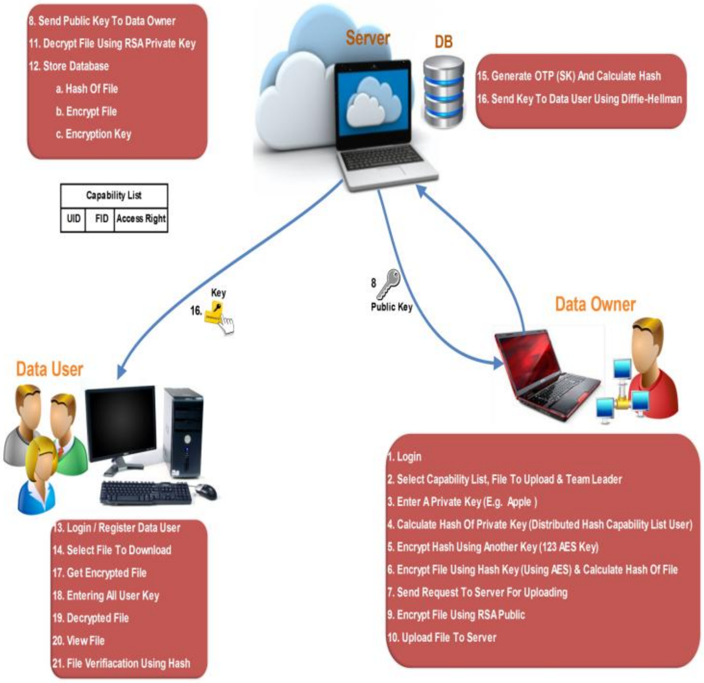
AES, RSA and DES comparisons does not confirm any of them as the best in all algorithms instead they all have different advantages and disadvantages. AES consumes less memory whereas DES takes less time and RSA will result in less output bytes. [4]

In Our method we have only used AES and RSA as per the need.

The remaining portion of this paper is organized as follows. In Section II, we review the related work. In Section III, we present the proposed scheme. In Section IV, we describe the future scope. Section V gives the performance analysis and finally in section VI we conclude our paper.

III. Proposed scheme

In this section we will be elaborating the entire scheme in detail. We have divided the steps of each entity to simplify the example. Better understand our model we will keep referring to the image/architecture diagram below.


8. Send Public Key To Data Owner
11. Decrypt File Using RSA Private Key
12. Store Database
a. Hash Of File
b. Encrypt File
c. Encryption Key
Server
DB
15. Generate OTP (SK) And Calculate Hash
16. Send Key To Data User Using Diffie-Hellman
Capability List
UID
FID
Access Right
Key
16.
8
Public Key
Data Owner
Data User
1. Login
2. Select Capability List, File To Upload & Team Leader
3. Enter A Private Key (E.g. Apple)
4. Calculate Hash Of Private Key (Distributed Hash Capability List User)
5. Encrypt Hash Using Another Key (123 AES Key)
6. Encrypt File Using Hash Key (Using AES) & Calculate Hash Of File
7. Send Request To Server For Uploading
9. Encrypt File Using RSA Public
10. Upload File To Server
13. Login / Register Data User
14. Select File To Download
17. Get Encrypted File
18. Entering All User Key
19. Decrypted File
20. View File
21. File Verifiacation Using Hash

We are taking an example of a company which uses cloud services. [2]

We divide the scheme in three parts.

1. Owner uploading the file securely.
2. Cloud server processing on the file.
3. End user getting the hands on the file by decrypting it and reconfirming file integrity.

In The First Part:

The first step will be registering or login on the server of Cloud Company. Here we will store all the information of the owner so as to authenticate the actual owner of the data.

The owner of data will select the file and will assign the name of team leader with it. Along with it we are also using the capability list similar to the approach in [2]

Then, the owner will enter a private key and will calculate the hash of that key (SHA). This Hash is again encrypted using AES key. Now the file is encrypted using the hash code (using AES) and a Hash of the file is generated. Then this file is ready to be uploaded on a server. But to double ensure file integrity we will encrypt the file using RSA algorithm using the public key sent by cloud server. And finally this file is uploaded on the cloud server. This whole part was majorly performed on the owner's side. Now we will see what the cloud server and the user will do to acquire the file safely.

In second part:

As the file is received by cloud server it will decrypt that file and now will have Hash of the file, the encrypted file itself and an encryption key. All of these will be stored on the database of the cloud server which we assume is secure.

In third part:

Now whenever the user needs that particular file he will have to follow a process.

At first he will authenticate himself by entering login/registration details. Then, He will select that file to download from the server. The OTP for that user will be generated which is the symmetric key and its hash will be calculated. Now this file is sent to the user using Diffie hellman algorithm to avoid man in middle attacks [2]

Now the user will have to enter not just his password but also minimum number of passwords will have to be entered by his team members. The work is still not done, now he will have to send a request to the team leader explicitly, to enter his part of the key as until and unless Team leader's part of the key is not entered the file cannot be decrypted.

If met with threshold and also the TL key is received the file can now be decrypted. But as cyber criminalists are not just about getting access to data, if they don't get access they may still try to destroy the data or at least change the data somehow. To recheck if any such thing has not happened with the file and the file is still the same the hash codes of the file at the cloud server is compared to the one which is bundled with the file thus maintaining file integrity. This scheme is a complete model for clod based industries and clod servers.

IV. SETUP AND IMPLEMENTATION

The setup is divided in three systems:

1. Cloud server.
2. Data owner.
3. User.

Cloud server-

The Cloud server stores the encrypted files in its database. The file that will be received from the Data Owner.

Note that we have performed two encryptions on the file at the Data Owner side. To transfer it from Data Owner side to Cloud Service Provider we have encrypted the already encrypted file using RSA Public key. Hence we are encrypting the file twice which increases the security. Even the cloud server cannot open the file.

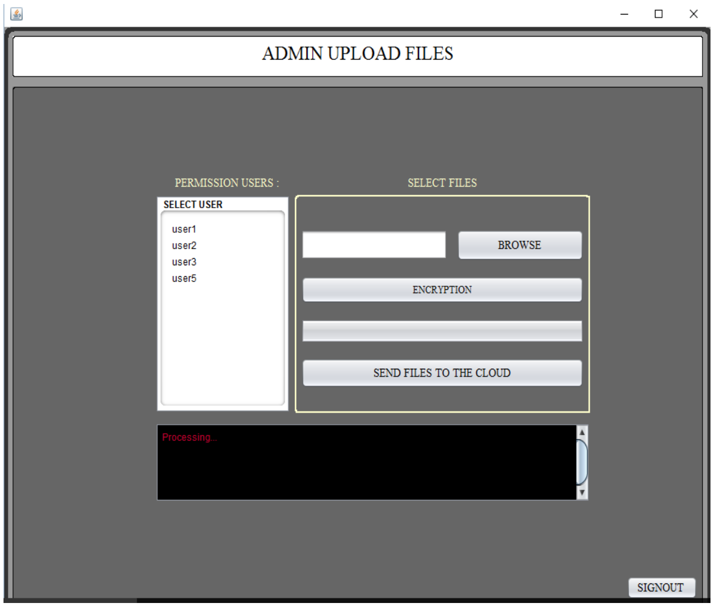
The Cloud Service Provider generates the One Time Passwords which are hashes of the password entered by the data owner and passes it to the Users to whom the file as being allotted.

We are deploying the Cloud Service Provider using GlassFish Server.

Data owner:

Data Owner is the system which wants to share a file over the cloud to a team. On the Data Owner System, the Data Owner selects the file he wants to upload. After selecting the file, the Data Owner enters a Private key. Hash is calculated of the Private Key. Enter one more key to encrypt the Hash key using AES. The file is encrypted using the Hash Key using AES. Data Owner selects the users to whom he wants to share the file. The encrypted file is encrypted again using RSA Public Key to upload it to the Cloud Server. After successful encryption the file is uploaded to the server

A screen shot of DO module:



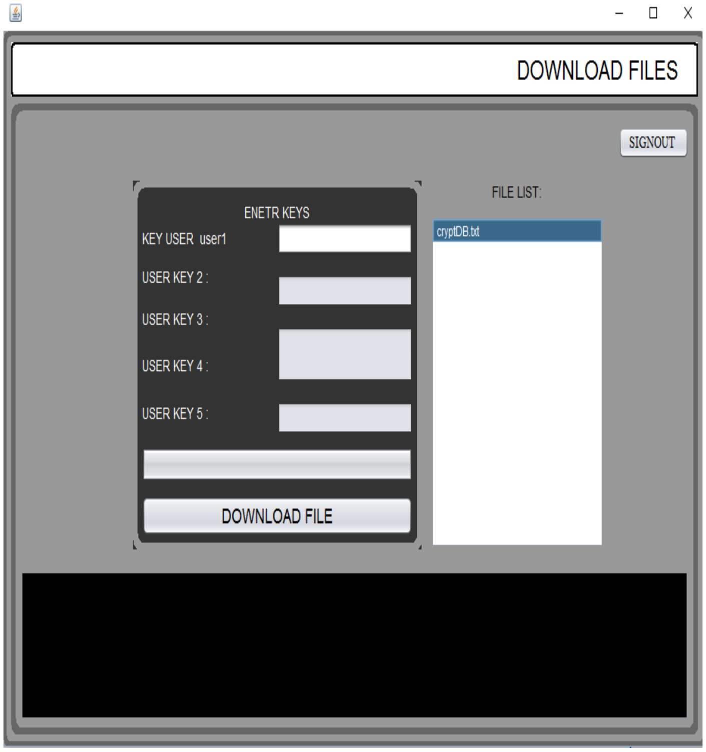
Processing...

User system:

Users are group of users in an organization. To access the file on the Cloud, the users are provided with on time passwords (OTP). These one time passwords are nothing but the Hash key generated which is broken down in to ‘n’ number of parts according the users selected by the group. Among these Users is a Team Leader whose OTP is essential for decrypting the file. According to the threshold set by the Data Owner, while decrypting at the Users end the threshold should be met as well as the Team Leader’s OTP should be entered.

If wrong keys are entered the file is still downloaded but in encrypted format. Hence, no use of it to an intruder.

A screenshot of user module:



V. PERFORMANCE

The Encryption algorithms AES and RSA do their job well. Double encryption of key results in a robust and extra secure environment for file upload and download. The hash comparison technique confirms data integrity. Team Leader based approach prohibits group attacks. As a single key is being divided and that to on demand division the key overhead is considerably reduced.

VI. FUTURE SCOPE

Even though AES and RSA are roust algorithms, they are famous and a lot of cyber criminals are trying to find loop holes in these algorithms. Designing new cryptographic algorithms has always been a future scope and it will still be a future scope. Although team leaders are chosen properly they can still join the malicious activities and hence finding a new approach that completely excludes this risk is still a job.

VII. CONCLUSION

This scheme represents a complete model for providing security to any industry who utilizes cloud as service. Not just it provides security but also reduces the number of cryptographic keys on cloud server there by benefiting both the sides the, server and the client. There is no effect of size of the files on the security the protection is still provided and keys are still reduce no matter how much large the file is. The Team Leader approach provides an edge to the whole process. The hashing and encrypting of data

Makes this scheme n extremely secure. This scheme will prohibit the malicious insiders to do any criminal activities as they cannot access the data without a proper permission and access rights.

REFERENCES

[1] John Harauz, Lori M. Kaufman, Bruce Potter “Data Security in the World of Cloud Computing” IEEE SECURITY & PRIVACY

[2]Sushil Kr Saroj, Sanjeev Kr Chauhan, Aravendra Kr Sharma, Sundaram Vats” Threshold Cryptography Based Data Security in Cloud Computing” 2015 IEEE International Conference on Computational Intelligence & Communication Technology

[3] Amanjot Kaur, Manisha “Bhardwaj HYBRID ENCRYPTION FOR CLOUD DATABASE SECURITY” INTERNATIONAL JOURNAL OF ENGINEERING SCIENCE & ADVANCED TECHNOLOGY Volume-2, Issue-3, 737 – 741

[4] Shashi Mehrotra Seth, Rajan Mishra” Comparative Analysis Of Encryption Algorithms For Data Communication” IJCST Vol. 2, Iss ue 2, June 2011 I S S N : 2 2 2 9 - 4 3 3 3