A Novel Technique to Improve Data Security in Cloud Architecture

Kritika Mehta, Akash Shrivastava
Department of Computer Science, DIT University, Dehradun, INDIA

Abstract

Cloud computing is a latest technology and is getting more popular day by day because of the services it offers – on demand computing, pay per use, storing, managing and accessing data etc. But there are some issues in cloud computing, the main one is security because every user store their useful data on the network as they want their data should be protected from any unauthorized access, any changes that is not done on user’s behalf. That is why various types of encryption techniques are being used like FDE and FHE. Though the various schemes have been proposed to solve the problem of key management, The third party scheme is mostly used to solve the problem of key sharing and key management! The third party auditing scheme will be at risk, if the security of third party is compromised or if the party is not authenticated. To solve this problem of third party, we have tried to design a new modal for key sharing and key management in fully Homomorphic Encryption scheme. In this paper, we have used the symmetric key agreement algorithm named Diffie Hellman, it is method of cryptographic key exchange which create session key between two parties who want to communicate with each other and HMAC for the data integrity, OTP(One Time Password) is created which provides more security. Due to this the problem of managing the keys is removed and data is more secured.

Keywords: OTP, HMAC, Diffie-Hellman, Cloud Security, FHE, FDE.

I. INTRODUCTION

Cloud computing provide the convenient environment and on demand network access to a shared pool of computing resources like servers, networks, applications, storage and services that can be rapidly released in an efficient way. The word cloud stands for internet, thus cloud computing means internet based computing. Cloud is a centralized database where many clients/organizations store their data and possibly modify data and retrieve data [7]. Cloud is a model where services are provided by CSP (cloud server provider) on pay per use base to user, means here client has to pay only for what he is using or being served. Cloud computing is a technique which provide a huge range of applications under different kind of topologies and every topology drives some new specialization. Even cloud service provider like Drop box could accidently allows accessing any user’s account without user knowledge. This would potentially lead to massive data breaches which are beyond user’s control [4]. To fortify the security for cloud computing most organizations adopt standard enterprise security solutions like firewall, IPS, and antivirus. Since users can now access cloud services from anywhere around the world. Some organizations may implement strong user authentication and access control solutions as a defense against identity frauds. Unfortunately, these solutions do not actually protect the user’s data in the cloud. The cloud computing model has three service job models and three set up models. The three service job models are:

1. Cloud software as a service
2. Cloud platform as a service
3. Cloud infrastructure as a service

The three set up models are:

1) Private cloud
2) Public cloud
3) Hybrid cloud

Cloud computing has particular characteristics that distinguish it from classical resource and service provisioning environments [12]:

1) Infinitely (more or less) scalable
2) Cost saving/less capital expenditure
3) Higher resource utilization
4) Business agility
5) Disaster recovery and backup
6) Device and location independence

While reducing up-front IT cost or capital expenditure is the one of the crucial reason for the adopting the cloud computing, there are also some other factors that encourage the various organizations for the adoption of cloud computing. Cloud computing migrate the location of resources to the cloud to cut down the costs associated with having too many resources, not using resources adequately and under-provisioning. It also reduces the time required to provision resources in minutes, allowing applications to quickly scale under-utilization both up and down, as the workload changes. Therefore, cloud computing is mainly well suited for applications with an inconsistent workload that practice hourly, daily, weekly or monthly changeability in exploitation of resources. One example of
such applications is online shops, which have to handle their peak loads at Deepawali time. Another example is university websites, which have to handle their peak loads during exam result time.

In traditional (i.e. non-cloud) environments, over provisioning and under-utilization can hardly be avoided [10].

Cloud Computing Security: Network security, information security and many other security types like the computer security together make the term “Cloud Security”. Because it consist all of the security mechanism given above. It gives the broad set of technologies, policies and controls that are used to secure the data and applications exist with the cloud computing environment [8]. It is not the product of computer security like anti-viruses and anti-spam’s. Security is the most concerning point to any service. External security or internal security required to each field. Only security ensures the privacy and integrity the cloud data. There are many security loopholes exist in the service. There are many types of security issues exist like DDOS, Man in the middle etc. Some security types include:

a) Outages: This term refers to the problem of the user where he/she is not able to access services due to the provider being down. Suppose there is some important business meeting and user require a document for the presentation and provider’s site is down. This might happens lot of times [8].

b) Data Loss: Due to lack of security data might be lost during uploading on cloud due to presence of malicious node [11].

c) Phishing: It is an e-mail fraud scam which is conducted with the help of network analysis flow tool to extract information from the server.

II. REVIEW OF LITERATURE

In this paper [1] they proposed different techniques and their merits and demerits like Message Authentication Code(MAC) which protect the data from integrity. The owner of any information verified the data integrity by recalculating the message authentication code of data received by others but recalculation is possible if the amount of data is very large. A hash tree is used for large files. Third party auditor is used to relieve the large data into small parts of maintenance and security. The proposed algorithm describes data integrity and dynamic data operations. They use encryption to ensuring the data integrity. Public key is also defined which is based on homomorphic authenticator. A hash function is used for proof of retrievability. The proposed algorithm has a main drawback that it require implementation of the higher resources cost. In this paper [2] Dynamic mobile token application is introduced. This is the application in mobile phones which is used to generate a code with the help of OTP (One Time Password). This OTP code is used only for one time to login session. In this paper, they describe one of the methods of OTP. There are two phases in it Registration phase and Login phase. User first register itself by fill credentials in the form and then enters to the Login phase. In login phase, OTP will generate for the login session. OTP is generated by three parameters: The current time, 4-digit PIN code and Init-secret. This code is valid for three minutes only. This ensures protection against eavesdroppers attack and man-in-middle attack. Hence, they prove OTP is very secure. In this paper [3] a design and architecture is proposed that can help to encrypt and decrypt the file at the user side which provides data security in both cases while user is at rest or is transferring data. In this paper they used the Rijndael Encryption Algorithm along with EAP-CHAP. This algorithm has five steps which need to be follow for the data security. The users are always concern about the privacy protection and security issues before storing their data on cloud. So in this the focus is on client side security in which only the authorized user can access the data. Even if some intruder (Unauthorized user) gets access of the data then the data will not be decrypt. Encryption must be done by the user to provide better security Algorithm. For this, Rijndael Encryption algorithm is used. In this paper [4], two techniques are discussed: Virtualization and Multi-tenancy which provides security about cloud computing. As data is organized by third party organizations that offer Saas and PaaS which is important for the security. So, Virtualization and Multi-tenancy techniques are used for the security purposes. Virtualization is a way of making a physical computer function as if it were two or more computers where each non-physical or virtualized. There are two types of virtualization: Full virtualization and Para virtualization and two architectures of virtualization: Hosted and Hypervisor architecture. Multi-tenancy is the ability to provide computing services to multiple customers by using a common infrastructure and code base. Multi-tenancy can be applied to different levels i.e. application level, middleware level, operating system, hardware level. Then security of virtualization and multi-tenancy has been discussed. In this paper [5] they discussed different issues related to cloud computing security. To protect cloud computing system and to prevent various attacks many security mechanisms have been developed. To improve the security of cloud computing new technologies has been developed by the researchers. Different types of attacks like SYN flood, malware injection, account hijacking are discussed in this paper. The main focus of this paper is on detecting and preventing SYN flood in cloud computing. The author developed two algorithm one detecting algorithm and one preventing algorithm. They will implement and test these algorithms on cloud computing.

III. APPROACHES USED

a) Diffie hellman

Diffie Hellman was the first public key algorithm or we can say that it is symmetric key agreement ever invented, in 1976. Diffie Hellman key agreement protocol is [6]:

1. It allows exchanging a secret key between two parties.

2. Exponential key agreement

3. Requires no prior secrets

Before establishing a symmetric key, the both the two parties need to choose two numbers n and p. Let n be a prime number and p be an integer. The Diffie Hellman Problem (DHP) is the problem of computing the value of
p^{ab}(\text{mod } n)\text{ from the known values of } p^a(\text{mod } n)\text{ and } p^b(\text{mod } n). The setup of Diffie-Hellman algorithm

1. Suppose that we have two parties Alice (Master) and Bob (Slave), they want to communicate to each other.
2. They do not want the eavesdropper to know their message.
3. Alice and Bob agree upon and make public two numbers n and p, where n is a prime number and p is a primitive root mod n. Anyone has access to these numbers.

Table 1: private computations

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choose a secret number a</td>
<td>Choose a secret number b</td>
</tr>
<tr>
<td>Compute M = p^a mod n</td>
<td>Compute S = p^b mod n</td>
</tr>
</tbody>
</table>

- Generated public values are exchanged.
- Alice sends M to Bob == M
- S= Bob sends S to Alice
- Alice calculate the number K= S^a = (p^a)^b (mod n).
- Bob calculate the number K=M^b = (p^b)^a (mod n).

Here Alice and Bob have the same key that is K=p^{ab} (mod n).

In the Diffie-Hellman algorithm if two parties, say, Master and Slave wishes to exchange data, both agree on a symmetric key. For encryption or decryption of the messages symmetric key is used. We knows that Diffie Hellman algorithm is used for only key agreement or key exchange, but it does not used for encryption or decryption. Before starting the communication, secure channel is established between both the parties [5]. Both parties select their own random number. On the basis of the selected random numbers, secure channel and shared key is established.

**Figure 1**: Diffie-Hellman Key exchange

Master computes:
\[ M = p^a \text{ mod } n \]
Slave computes:
\[ S = p^b \text{ mod } n \]

Now both master and slave exchange their private keys such as ‘M’ and ‘S’. After getting ‘M’ and ‘S’, master and slave calculates the secret keys such as K1, K2.

From S, master computes:
\[ K1 = S^a \text{ mod } n \]
From M, slave computes:
\[ K2 = M^b \text{ mod } n \]

If both master and slave calculate same values of K1 and K2, then secure channel is established between them. The combination of K1 and K2 becomes the shared symmetric key between master and slave.

To encrypt the messages, they used the public key or shared key (K) of both parties. For decryption of messages private key of both parties which is randomly chosen by the users i.e. ‘a’ and ‘b’ are used [16].

b) One Time Password: Password is used for authentication by all the business and organization. Moreover Static passwords have many limitations. Password can be get hacked. Lackadaisical employee may note down passwords somewhere, system with saved passwords may be used by various users or a malicious user may reset all passwords just to create havoc. So it is very useful to use dynamic password i.e. one time password [10]. Dynamic passwords are more secure as compared to static. There is no need to write down these passwords and remember these passwords. For each login session each time a new password is generated. One time passwords are more reliable and user friendly as well for authentication. OTP generation can be done by various OTP generation algorithms for generating strings of passwords. OTP ensures safety. This leads to authenticating them again and again over the period of time for each login session. To avoid the overhead we can use OTP for multi cloud environment.

IV. PROPOSED METHODOLOGY

There are many encryption algorithms to provide security to the cloud. “Fully Homomorphic” is more reliable. It gives more privacy and security as compare to scheme of “Full Disk Encryption”. The main problem which is there in Fully Homomorphic Encryption is a key storage, key management, Access control and Data Aggregation list maintaining. To solve problem of Key
management, Key Sharing various schemes have been proposed in last years. The various security attacks are possible in these schemes. The third party auditor is the scheme for key management and key sharing. The third party auditing scheme will be failed, if the third party’s security is compromised or of the third party will be malicious. To solve this problem, In this thesis we will work on to design new model for key sharing and key management in fully Homomorphic Encryption scheme. In this work, we find that fully homomorphic encryption technique is more efficient than full disk encryption. But the main problem exists in fully homomorphic encryption is of key management and key sharing which reduces the reliability of the scheme. For key management and key sharing, enhancement has been proposed in the encryption scheme and enhancement is based on Diffie-hellman algorithm and HMAC and OTP is generated on the basis of secret key generated from diffie-hellman algorithm. This algorithm create session key between user and cloud. Each time new key is generated between two before communication selected node suppose user1

1. Login
2. Key generation
   2.1 Enter prime numbers
   2.2 Enter random numbers by client and cloud service provider
   2.3 Secret key generation and secure channel establishment
3. OTP (One Time Password) generation
   3.1 Cloud server will set count1=0, count2=0...count5=0 for respective user at its side.
   3.2 Cloud Server will request for the OTP from user
   3.3 user1 enter (secret key+count) as OTP
   3.3 server match it because server knows both secret key and count of each user.
   3.3.1: count1++; // so for user 1 it will be count1=1; for remainig user their count will be still 0;
   3.3.2 if ( secret_key+count(x) == secret_key+count(y))
       { Access granted;
         display message by server : print ("please enter the operation");
       } else { display message by server: print("wrong password, your login number is count1");
   4.4 client will enter the operation using HMAC digest
   4.4.1 : hmac(already generated secret key || v, file1,ver1 || sha1)
   { if(ope==v)
         { server will check the file name and version;
         ...
   }

V. EXPERIMENTAL RESULTS
The whole scenario has been implemented on MATLAB tool.

Figure 2 : Comparison Graph
As shown in figure 2, the comparison between previous and proposed approach is shown in terms of delay. The delay in previous technique is increasing, when numbers of exchange messages are increased. In the proposed approach the delay is less due to increasing the number of message.

Figure 3: Comparison with Diffie Hellman in terms of used bytes
As shown in figure 1.3, the comparison between previous and proposed approach is shown in terms of used bytes. The used byte in previous technique is increasing, when numbers of exchange messages are increased. In the proposed approach the data usage is less as compared to existing technique.

VI. CONCLUSION

Cloud computing is the environment which provides on-demand & convenient access of the network to a computing resources like storage, servers, applications, networks and the other services which can be released minimum efficiency way. In this user can store their data and use different services and pay according to those services. The main factor is security that how we can store our data while storing into the cloud. In this thesis, we reviewed two most popular techniques for cloud data encryption. These techniques are full disk encryption and fully homomorphic encryption. In this work, we find that fully homomorphic encryption technique is more efficient than full disk encryption. But the main problem exists in fully homomorphic encryption is of key management and key sharing which reduces the reliability of the scheme. For key management and key sharing, enhancement has been proposed in the encryption scheme and enhancement is based on Diffie-hellman algorithm and HMAC and OTP is generated on the basis of secret key generated from diffie-hellman algorithm. This algorithm creates session key between user and cloud. Each time new key is generated between two before communication. This reduces the time takes place in management and sharing of keys and secure channel is established between both i.e. user and the cloud service provider. The simulation shows that proposed enhancement is more efficient and reliable than the existing one.

REFERENCES


