Dual Cryptography Based Data Security in Cloud Computing

[1] Nisha Chaube, [2] Dushyant Singh, [3] Ayush Kr. Shrivastava, [4] Sachin Kathuria
[1][2][3] B.Tech Student [4] Assistant Professor
[1]nishachaube1@gmail.com, [2] dushyant.siddhu79@gmail.com, [3] ashrivastava288@gmail.com,
[4] kathuriasachins@gmail.com

***Abstract:* - Cloud computing is a new paradigm of distributed computing which provides data and other resources to users on demand over the network. Cloud applications offer reliability, utility, scalability and availability. A lot of cloud servers provide cloud services. The cloud provides a number of services like Software as a service (SaaS), Platform as a service (PaaS), and Infrastructure as a service (IaaS). Each service has their own associated security issue. With this comes a number of challenges in its implementation. This paper focuses on the major challenge on Security & Privacy related with Cloud Computing. The issues related to privacy have been addressed along with a proposal of a new implementation of cloud computing architecture. We have used segmentation technique to apply different encryption algorithms on different segments on the basis of odd even index values of segments.**

***Keywords*- cryptography, index-hash mapping table, MD5 hashing technique, multi-cloud, rijndael algorithm, serpent algorithm..**

I. INTRODUCTION

Cloud computing is one of the fastest growing technologies in the field of computation and storage of data and files. Cloud computing refers to manipulating, configuring and accessing the resources remotely. It offers online data storage, infrastructure and application on demand of the client with minimal management effort. It is a kind of Internet-based computing that provides shared processing resources and data to computers and other devices on demand. There are three basic services models[1] - infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS), working behind the scene making the cloud computing feasible and accessible to the users. There are several attributes of cloud computing which allows users to deploy the cloud services. These are as follows:

1) ***Reliability***: Cloud computing is a reliable and consistent option when there is a managed service platform.

2) Scalability: It provides the ability to scale the bandwidth and storage space of the system.

3) Elasticity: Users can rapidly increase and decrease their computing resources as needed.

4) On-demand service: Users are provided the services according to their demand and are allowed to pay for only the resources they actually use for a particular time frame.

Along with all the advantages of the cloud services, the main area of concern in the field of cloud computing is security. Cloud Computing stores the data in the open environment at external servers which makes the data prone to potential attacks. The data owners feel unsafe to store their data on cloud. The cloud provider should ensure that the architecture and the infrastructure used by them is secure and the data and applications of the client are not compromised. Even at the time of transmission of data it is required that the data is not prone to any kind of active or passive attack.

One way of resolving this issue is encryption. The main concern of encryption is to provide-

1) Confidentiality
2) Data integrity
3) Authentication
4) Non Repudiation

There are various algorithms [2] used for the encryption. These are classified into two categories-

***1) Symmetric Key Algorithms*:** Symmetric key algorithms use single shared key to encrypt as well to decrypt the data. The sender encrypts the data using a key and then send the cipher text to the receiver. The receiver uses the same shared key to decrypt the data.

2) Asymmetric Key Algorithms: Asymmetric key algorithms use two different keys for encryption and decryption processes. These keys are known as public key and private key. The sender encrypt the data using the public key of receiver and send the cipher text to the receiver. The receiver then decrypt the cipher text using its private key. As compared to symmetric key algorithms, asymmetric key algorithms are more secured.

The encryption can be done at server side as well as at client side.

1) Server side encryption: When data is sent to the service provider then the data is encrypted first and then stored at cloud. But this scheme of encryption is more vulnerable as data can be attacked in the

way to the service provider which will result in loss of integrity of data.

2) Client side encryption : Another way of maintaining data integrity is that the data should be encrypted at client side before sending it to the service provider

There are several techniques of encryption which are proposed to ensure the security requirement of data but they are still prone to security breach. To address this issue we have proposed a new scheme of encryption of data at client side which is highly secure and is meant for storage of highly confidential data.

II. RELATED WORK

Data owner depends upon Cloud Computing for various computing applications and data storage. Here, the security of data is wholly in the hands of Cloud Service Provider. The cloud service provider should ensure that proper mechanism is applied for the security of the cloud storage. Cloud service providers ensure the security at server side and try to claim that they have taken efficient and appropriate measures to secure their cloud storage. But there have been many attacks on various cloud service providers so the data security at client side becomes equally important. When data owners outsource the data to cloud, data confidentiality and data integrity are the two main factors which should we kept in mind. So to ensure security at client side various attempts are made to design various schemes. In [3], Tania Gaur and Divya Sharma proposed a scheme of applying cryptographic technique of encrypting the data at client side before outsourcing it to the cloud. They make use of the concept of the Diffie-Hellman algorithm to generate a shared key which is used for encryption and decryption. The used AES (Advanced Encryption Standard) algorithm to encrypt the data using the key generated by the Diffie-Hellman algorithm. This data is then uploaded to the cloud and the key is shared at network through Third Party Auditor (TPA). In[4], a Virtual Machine is provided through the means of Software as a Service which encrypt the data using RSA algorithm and SHA is used to create message digest of this encrypted data is sent to Cloud using Third Party Auditor (TPA). The scheme proposed in [6][7][8], also shows the use of third party auditor to store the data to the cloud. But when we deal with highly confidential data, we cannot rely on Third Party Auditor (TPA). These schemes are vulnerable when Third Party Auditor (TPA) is not trustworthy. In [5], both symmetric and asymmetric encryption techniques are used. The data is encrypted using Blowfish algorithm and then the encrypted data is converted to message digest using MD5 technique. In [7], the proposed scheme uses the Byte Rotation Encryption Algorithm (BREA) which decomposes the data into blocks of 16 bytes and then monoalphbetic substitution concept in followed and byte rotation scheme is used [10] which describes various encryption technique such as Advanced Encryption Standard (AES), Hybrid Vigenere Caesar Cipher Encryption (HVCCE), Fully Homomorphic Encryption, Hierarchical Identity Based Encryption (Hibe). In [12], two level of security is applied using DES and RSA. At first level of encryption DES is used to encrypt data and then this encrypted data is again encrypted using RSA algorithm. Comparative study of [12][13][14] shows the use of various symmetric and asymmetric algorithm used for encryption the data. These proposed schemes either use single encryption algorithm or uses single cloud to store the data. In the worst case, if the attacker is able to guess the algorithm used for encryption by repeatedly analysing the pattern of the cipher text then the confidential data will get exposed. So, when we deal with highly confidential data, the high level of security at client side becomes highly important.

III. MODEL AND ASSUMPTIONS

We have assumed that the model is composed of three entities namely, Data owner (DO), Cloud service provider (CSP) and the User. We are assuming that the cloud is highly secure at its architecture. In order to exchange data and store it on cloud, the user needs to get itself registered at DO. As a result DO shares encryption keys, algorithms, index-hash mapping table and other relevant information. Only upon successful authentication can the data be stored and retrieved from the cloud.

To ensure a secure communication we have used the concept of application of two encryption algorithms based on their even odd index position. We have also proposed making an index-hash mapping table to ensure that the segments can be rearranged at the time of decryption. The data segments are encrypted with the hash values before sending it to the cloud. This increases the security of the encrypted data segments. Also the strategy to employ encryption is changed here. We are applying encryption on individual segments separately. This feature brings in enhanced security. The time happens to be a slight concern. Since the data is assumed to be highly confidential and important, a time trade off can be accepted at the expense of high level of security obtained. Even after capturing a segment or two, the hacker will not be able to identify the encryption technique used in them. It becomes very difficult to decrypt the entire message. The data is stored securely on multi-cloud. Here by multi cloud, we mean that the single cloud service provider has many sub clouds at scattered places. The data on multi-cloud is changed dynamically at regular intervals to ensure that the security of data is not breached at any cost.

IV. PROPOSED SCHEME

In this section, we present a complete model for secure communication between different entities and secure access to data. There are four algorithms in the proposed scheme. Algorithm 1 describes the procedure to break the file into segments. Algorithm 2 describes the technique to calculate the hash values of individual segments using MD5[15] hashing technique. Algorithm 3 describes the application of two different encryption algorithms i.e. Rijndael[16] and Serpent[16] on segments by finding odd even index values. Algorithm 4 describes secure communication of data between DO, CSP and the storage at cloud. Lastly, the reverse of above procedure occurs after checking the user's authorization to retrieve a file.

Our algorithm accepts a data of any size and then breaks the data file into segments. We have used a ceiling function of 2n to break the data. Once the data is broken into segments, an index-hash mapping table is created. Hash value corresponding to every index value is calculated using MD5 hashing technique. This is then stored in the table by Data owner and shared with the user. The odd index valued data segments are encrypted using Rijndael algorithm and the even index valued data segments are encrypted using Serpent algorithm. We have used the two best symmetric key algorithms to ensure high level of security. If the hacker happens to capture a data segment then he might not be able to decrypt the entire message as it is only a part of it. Also it would be difficult for him to decrypt as he might not know which of the two algorithms were used to encrypt it.

The segments after being encrypted individually are sent to the cloud for secure storage. Concept of multi- cloud storage has been used to store the data on cloud. Even on the multi cloud, we have suggested dynamic position changes of the segments (inter and intra cloud).

When the user wishes to fetch the data from the cloud. He will be required to send the hash values of the segments to the cloud. Cloud service provider sends the data segments corresponding to the demanded hash values to the user. Upon receiving the segments the user rearranges the segments using the index-hash values table. Then the appropriate decryption algorithm is applied.

Algorithm 1: Procedure to be followed to break the file

- ❖ **Step 1:** DO uploads the data file
- ❖ **Step 2:** The uploaded file is broken into segments using the formula ceiling f(2n) Where n corresponds to the number of segments.

Algorithm 2: Procedure to apply MD5 algorithm

❖ ***Step 1: Append padding bits.***

The input message is "padded" (extended) so that its length (in bits) equals to 448 mod 512. Padding is always performed, even if the length of the message is already 448 mod 512. Padding is performed as follows: a single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits of the padded message becomes congruent to 448 mod 512. At least one bit and at most 512 bits are appended.

❖ ***Step 2: Append length.***

A 64-bit representation of the length of the message is appended to the result of step1. If the length of the message is greater than 2^64, only the low-order 64 bits will be used. The resulting message (after padding with bits and with b) has a length that is an exact multiple of 512 bits. The input message will have a length that is an exact multiple of 16 (32-bit) words.

❖ ***Step 3: Initialize MD buffer***

A four-word buffer (A, B, C, D) is used to compute the message digest. Each of A, B, C, D is a 32-bit register.

❖ ***Step 4: Process message in 16-word blocks***

Four functions will be defined such that each function takes an input of three 32-bit words and produces a 32-bit word output.

Algorithm 3: Application of encryption algorithms

- ❖ ***Step 1:*** An index-hash mapping table is created which maps individual index values with the hash values obtained after applying MD5 hashing algorithm.
- ❖ ***Step 2*:** Check if index value is odd

```
        Apply Rijndael algorithm
        Else
        Apply Serpent algorithm
```

- ***Rijndael algorithm***

```
Rijndael(State,CipherKey) {
KeyExpansion(CipherKey,ExpandedKey);
AddRoundKey(State,ExpandedKey); For( i=1 ; i Final
Round(State,ExpandedKey + Nb*Nr);
}
```

And the round function is defined as:

```
Round(State,RoundKey) {
ByteSub(State); ShiftRow(State); MixColumn(State);
AddRoundKey(State,RoundKey);
}
```

- ***Serpent Algorithm***

The cipher consists simply of 32 rounds. The plaintext becomes the first intermediate data B0 = P, after which the 32 rounds are applied, where each round i ∈{0,...,31} consists of three operations:

- ***Step 1: Key Mixing***: At each round, a 128-bit subkey Ki is exclusive or'ed with the current intermediate data Bi.

- ***Step 2: S-Boxes***: The 128-bit combination of input and key is considered as four 32-bit words. The S-box, which is implemented as a sequence of logical operations (as it would be in hardware) is applied to these four words, and the result is four output words. The CPU is thus employed to execute the 32 copies of the S-box simultaneously, resulting with Si (Bi $\oplus$Ki).

- ***Step 3: Linear Transformation***: The 32 bits in each of the output words are linearly mixed, by $X0,X1,X2,X3 := Si(Bi \oplus Ki)$

$X0 := X0 <<< 13$ $X2 := X2 <<< 3$ $X1 := X1 \oplus X0 \oplus X2$ $X3 := X3 \oplus X2 \oplus (X0 << 3)$ $X1 := X1 <<< 1$ $X3 := X3 <<< 7$ $X0 := X0 \oplus X1 \oplus X3$ $X2 := X2 \oplus X3 \oplus (X1 << 7)$ $X0 := X0 <<< 5$ $X2 := X2 <<< 22$ $Bi+1 := X0,X1,X2,X3$

Where <<< denotes rotation, and << denotes shift. In the last round, this linear transformation is replaced by an additional key mixing: $B32 := S7(B31 \oplus K31) \oplus K32$

Algorithm 4: Secure communication of data between DO and CSP

- ***Step 1:*** Send individual segments on the cloud randomly.
- ***Step 2:*** The segments are shuffled both inter-cloud and intra-cloud
- ***Step 3:*** During retrieval, the reverse of the above procedure occurs.

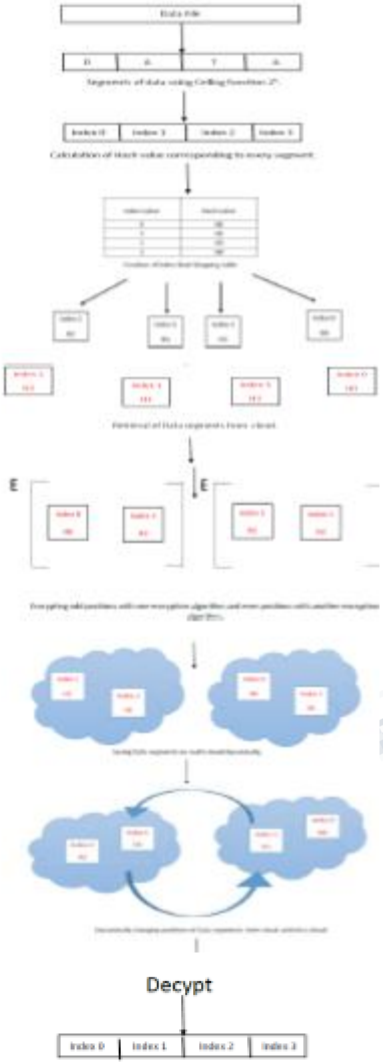

Fig.3. Flow of proposed architecture

V. CONCLUSION

In this paper, we presented a new approach which provides security for data outsourced at CSP. By employing the dual cryptography after segmentation at the user side, we protect outsourced data from outsider's attack. Since, DO stores its data at CSP in encrypted form and, keys are known only to DO and respected users group, data confidentiality is ensured. Segmentation provides an extra edge of security. To ensure easy retrieval of outsourced data, the scheme has used index-hash mapping table. Public key cryptography and MD5 ensure the entity authentication and data integrity respectively. The concept of multi-cloud provides additional security to the data stored at CSP. Within the CSP the data is dynamically shuffled both inter-cloud and intra-cloud. Public key cryptography with segmentation has protected the data from outsiders in our approach. The security has improved highly in the proposed scheme.

REFERENCES

[1]. G. Murugaboopathi, C.Chandravathy and P. Vinoth Kumar on Study on Cloud Computing and Security Approaches.

[2]. T. P.Wasnik, Vishal S. Patil, Sushant A. Patinge, Sachin R. Dave, Gaurav J. Sayasikamal on Cryptography as an Instrument to Network Security.

[3]. Tania Gaur and Divysharma (2016) on A Secure and Efficient Client-Side Encryption Scheme in Cloud Computing

[4]. Abhishek Mohta, Ravi Kant Sahu and Lalit Kumar Awasthi (2012) on Robust Data Security for Cloud while using Third Party Auditor, International Journal of Advanced Research in Computer Science and Software Engineering ISSN: 2277 128X.

[5]. Dr. Chander Kant and Yogesh sharma (2013) on Enhanced Security Architecture for Cloud Data Security, International Journal of Advanced Research in Computer Science and Sofware Engineering ISSN: 2277 128X.

[6]. Surendra Singh Rathod and Anand Rajawat (2015)