A Three-Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

Kavyashree M B[1], Shoba M[2], Nagashree C[3], Nischitha D Raj[4]

[1,3,4] UG Scholar, Information Science and Engineering Department, Sri Venkateshwara College of Engineering, Bengaluru, Karnataka, India
[2]Professor & Head, Information Science and Engineering Department, Sri Venkateshwara College of Engineering, Bengaluru, Karnataka, India

---***---

Abstract - *In recent years, which evidence the event of cloud computing technology with growth of unstructured data, cloud storage technology gets more attractive and development. Within the current storage scheme, user's data is stored in cloud servers where users lose the control over their own data and will face privacy protection risk. Moreover, privacy protection source is supported on encryption technology, these methods can't avoid attack from the insiders of cloud servers. To overcome this problem, we propose three-layer storage framework based on fog computing which take care of cloud storage and protect the privacy of data. However, Hash-Solomon code algorithm is meant to divide data into different parts where we are going to upload a small part of data in local machine and fog server to shield the privacy. Hash-Solomon code algorithm relies on computational intelligence, compute the distribution stored in cloud, fog server and local machine respectively.*

Key Words: Cloud computing, fog computing, privacy protection and cloud storage.

1. INTRODUCTION

Computer technology has developed rapidly since 21st century. Cloud computing is an on-demand technology, was first proposed by San Jose and as a great attraction from different sectors of society. There are cloud-based technologies derived from cloud computing, cloud storage is an important part.

Due to the rapid development of network, the user's data is rising geometrically and by the capacity of the local machine user's requirements can be satisfied any more. So, people try to find new methods to store their data to more powerful storage capacity, where number of user's select cloud storage. Storing data on cloud storage technology will become wide in future years and also provide data storage and management services. Network technology and distributed file system technology, cloud storage provides a large number of different storage devices to work together. Lot of companies provide a variety of cloud storage services, such as Google drive, iCloud etc, these companies provide large storage capacity and various services which leads to their success in attracting human subscribers. Moreover, cloud storage services exist a lot of security problems such privacy problem is significant among those security issues.

Here, user uploads the data directly to the cloud server subsequently the cloud server provider (CSP) will take place of user to manage the data. CSP can freely search and access the data stored in the cloud meanwhile attackers can also attack CSP server to obtain the user's data. So, the above two cases make users data loss and information leakage. Traditional secure cloud storage solutions are usually focusing on data encryption these methods can eliminate most part of the problems but all of these solutions cannot solve the internal attack. To overcome this, we propose a TLS scheme based on fog computing model and also design a Hash-Solomon code based on Reed-Solomon code. Based on cloud computing, fog computing is an extended computing model which is composed of a lot of fog nodes which have a storage capacity and processing capability. In our scheme, user's data is split into three parts and save them separately in the cloud server, fog server and the user's local machine. Depending on the property of the Hash-Solomon code, which ensure the original data cannot be recovered by partial data and it produce a portion of redundant data blocks used in decoding procedure. With increasing the number of redundant blocks can increase the reliability of the storage, it also results in additional data storage. Hash-Solomon code needs complex calculation, which is assisted with computational Intelligence (CI). In our scheme, we take the advantages of CI to do some calculation works in fog layer. Compared to traditional methods, our scheme can provide a higher privacy protection from interior, especially from CSPs.

2. SECURE CLOUD STORAGE BASED ON FOG COMPUTING

In SES 2006 (search engine strategies) San Jose cloud computing was first proposed and defined by NIST. By the rapid development of cloud computing technology, wherein now we can divide the data into fog server and local machine to preserve and protect our data in cloud storage with increased advantage.

2.1 Fog Computing

Fog computing is a transferable computing architecture here data is processed and stored between source of the origin and cloud infrastructure. More use of Internet of things (IOT) devices is primary motivation for fog computing wherein large amount of data is generated from an ever-expanding array of devices. In other words, we can also say fog computing is extended form of cloud computing on the edges of network. However, fog and cloud computing are similar data, computation, storage and api services the only difference lies in decentralization.

Features such as processing of huge amount of data locally, freely portable, operate on-premise and can be installed on heterogeneous hardware makes fog computing tremendously suitable for time and allocation sensitive applications. Fog computing model aims to give scalable solution for any issues. Challenges like exploding data velocity, variety and volume can be solved by fog computing offering cloud computing to handle the huge set of data generated daily through IOT.

2.2 Three-Layer Privacy Preserving Cloud Storage Scheme Based on Fog Computing Model

In current storage scheme user data is completely stored in cloud servers. However, with tremendous increase of unstructured data cloud storage gets more valued and demanded for better development. As traditional privacy protection methods and objectives are based on encryption technology, these cannot handle any hacking attacks from inner cloud server. So, to solve this issue, we put forward a three-layer privacy preserving storage framework which is based on fog computing. In this paper we divide the user's data into three parts based on size during encoding. The three layers are as such cloud server, fog server, local machine, where each of the layer have pair of key information for confidentiality. Therefore, hackers cannot get complete data as it will be divided into three, they will not be knowing where the other parts are stored if they somehow get with one part. So, we are giving the input files, the outcome will be enhanced security of the given input files by using the reed Solomon algorithm.

As shown in fig 1 using three-layer framework which consists of cloud server, fog server and local machine where largest amount of data i.e. almost 95% of data is stored in cloud, next portion which is 4% of data is stored in fog left over 1% in locality. While restoring say downloading whole data is downloaded in the locality.

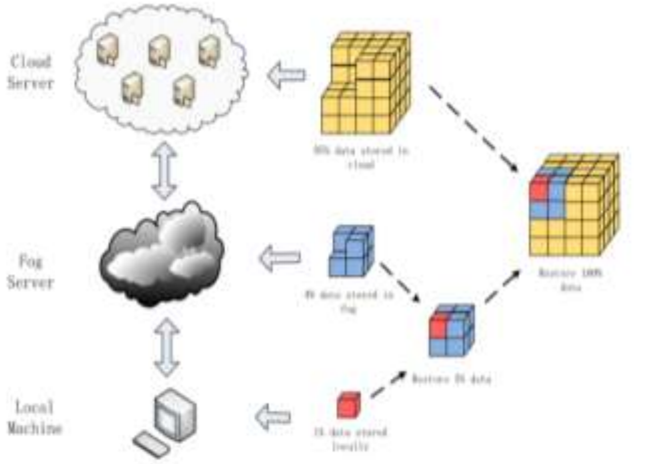


Fig-1: Representation on three-layer framework.

According to Reed-Solomon code the whole framework works where a buffer size is created for total number of file size initialized, whatever data is present is read and stored in byte form. Most likely it works through matrix particularly Vandermonde matrix. Vandermonde matric which is specifically used for geometric progression for all indices in each row. Unique property of this matrix is it does not have a property where data shards can be changed after encoding which is also termed as invertible. Data of less than 256 shards are considered if it extends there pops an exception. We also use an identical matrix called matrix top which is an inverse of top square of matrix. Adding on comes the parity were when checked during looping, if and only if parity is correct it will upload the data and encode.

The flow chart below represents the process control flow where to login registration should be done later once group manager activates our account then the authentication can be achieved.

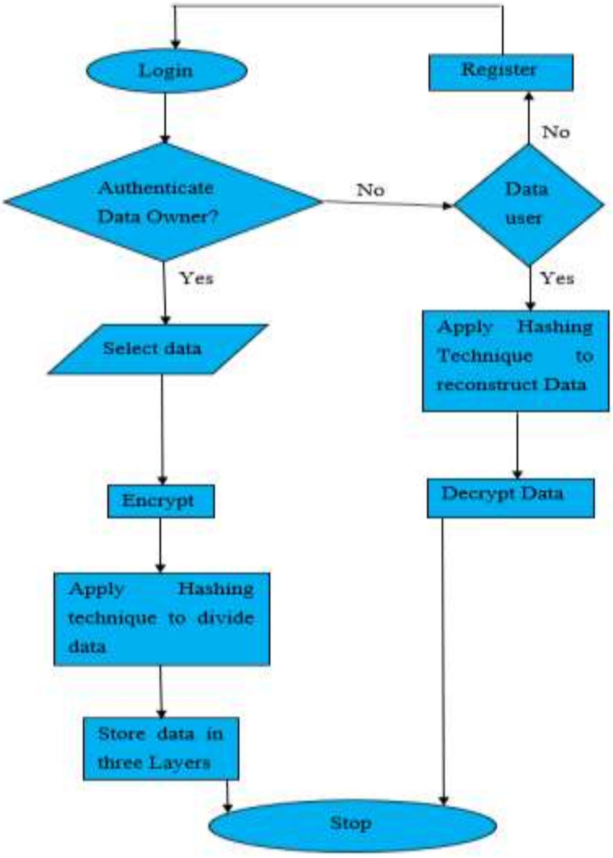

Login
Register
No
Authenticate Data Owner?
No
Data user
Yes
Yes
Select data
Apply Hashing Technique to reconstruct Data
Encrypt
Decrypt Data
Apply Hashing technique to divide data
Store data in three Layers
Stop

Fig-2: The process control flow

2.3 Implementation Detail of Workflow

1. **Stored Procedure:**

Data encoded by Hash-Solomon code is stored into three layers framework which consists of cloud server, fog server and local machine where largest amount of data i.e. almost 95% of data is stored in cloud, next portion which is 4% of data is stored in fog left over 1% in locality.

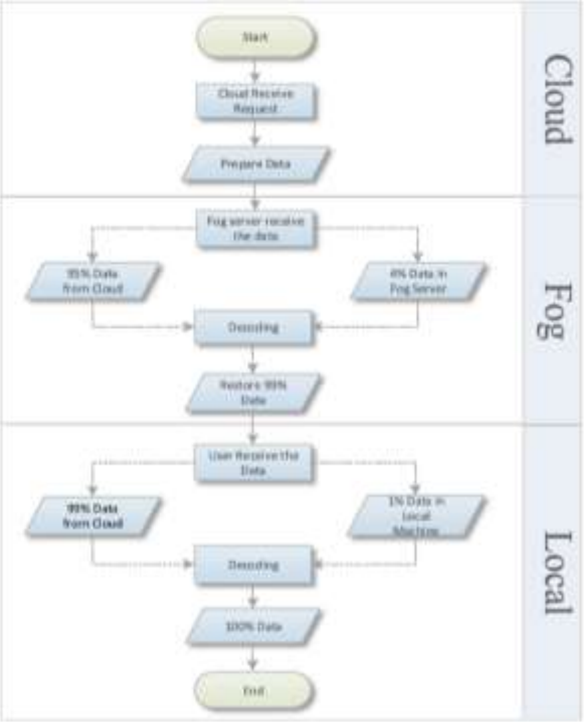


Fig-3: Stored procedure diagram

2. **Download Procedure:**

The data is integrated when the user requests cloud server which is in different server.

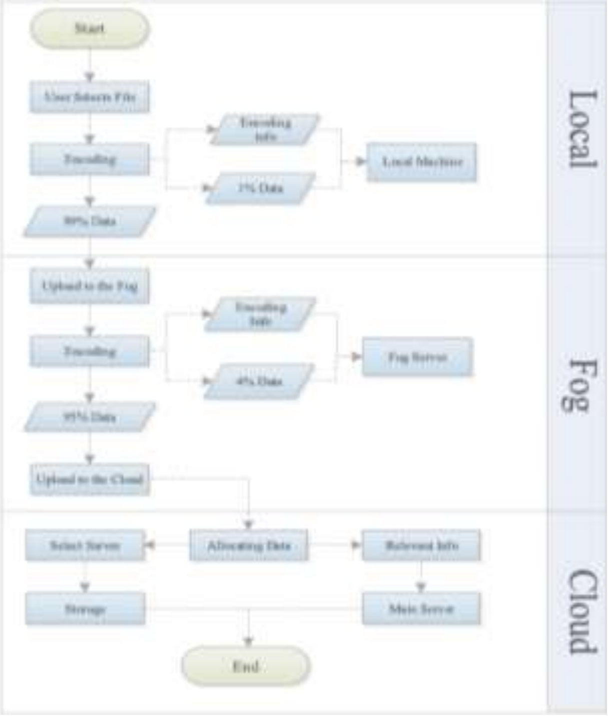
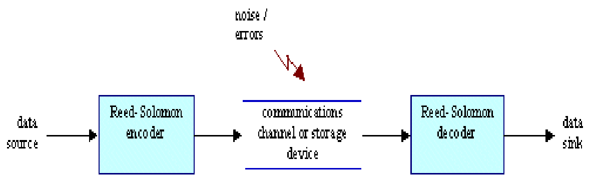


Fig-4: Download procedure diagram

2.4 Theoretical Safety Analysis

1. Reed Solomon Code:

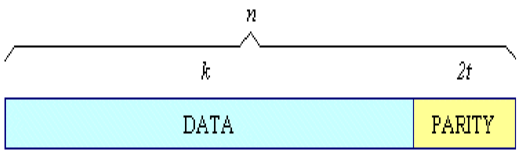
Reed Solomon codes are used to correct the block-based error with a high range of applications in digital communications and storage. It also used to correct the errors in many systems includes: Storage devices, Wireless or mobile communications, Satellite communications and Digital television.
A typical system is shown here:



The reed Solomon encoder blocks of digital data and adds more redundant bits. Error will occur when the transmission in process or storage for number reasons (noise, scratches on CD). The reed Solomon decoder will process each and every block and tries to correct errors and it tries to recover the original data. The number and the types of errors is corrected only by the characteristics of the reed Solomon code.

2. Properties of reed Solomon code:

Reed Solomon codes are the subset of BCH codes and they are also linear block codes. A reed Solomon code is determined as RS (n, k) with s- bit symbols. The encoder takes the many number of K data symbols of S bits and puts parity symbols to make an n symbol code word. There are n-k parity symbols of S bits. A reed Solomon decoder can correct up to the t symbols it contains errors in a code word, where 2t=n-k.



Because the unchanged of the data is left so the parity symbols are appended in the typical reed Solomon code word.

Example: A Reed-Solomon code is RS (255,223) with 8-bit symbols. Every code word contains 255 code word bytes, of which 223 bytes are data and 32 bytes are parity. For this code:

n = 255, k = 223, s = 8

2t = 32, t = 16

In the code word the any 16 bytes error can be decoded. The code word can be automatically corrected the error up to 16 bytes.

Given a symbol size s, the maximum code word length (n) for a Reed-Solomon code is **n = 2^s – 1**

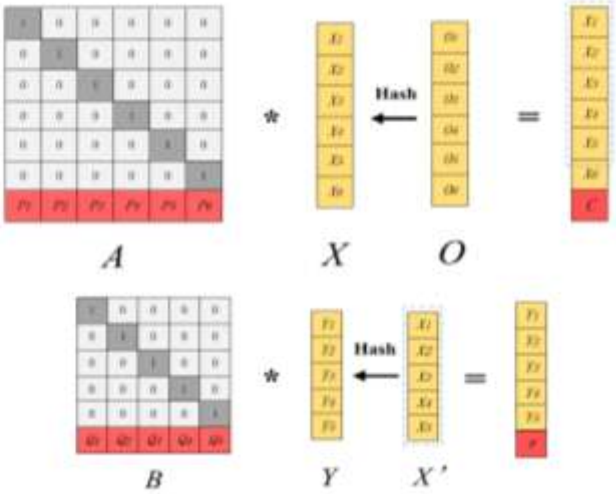


Fig-5: Diagram of download procedure (Hash Transformation)

Here we prepare to save X1 to X5 in the cloud server and fog server then X6 will be stored in the C in the local machine. To get the file matrix of Y we will do hash transform on X. Later we multiply the transformed matrix Y by the encoding matrix B. At last, we store Y1 to Y2 in the cloud server and Y5 will be stored in the R in the fog server. This encoding matrix usually consists of an identify matrix or a Cauchy matrix. The hash Solomon code has the following properties:

We must have at least K data blocks so that we can able to recover the original data combining with encoding matrix. But once the number of data blocks is less than K, it cannot be recovered by the data. Using the above properties in this project after each data blocks encoding the data will be stored in a higher server. And it should be less than K parts of the data blocks and the remaining parts of the data will be stored in the lower server.

In this the cloud server, fog server and local machine will automatically divide the data and stores certain percentage of the data blocks. The hacker cannot hack the data using the single server's data. Even though the attacker is brilliant enough if he steals the data from the two servers, he will get more than K parts of the data blocks. Even after getting the two servers data blocks he will not get the users information because the data blocks will be in the encoding matrix.

If he tries to crack the encoding matrix the M and K value will be in very large values, so it is impossible to crack the encoding matrix in theory. But using encoding technology we cannot ensure the privacy of the data blocks especially for document file. For example: a document is encoded but each part of the data blocks is still containing the information of the documents. So, we use a hash transform before encoding to the original data and then save the hash information in the user's local machine.

Original transform vs. Hash Transform

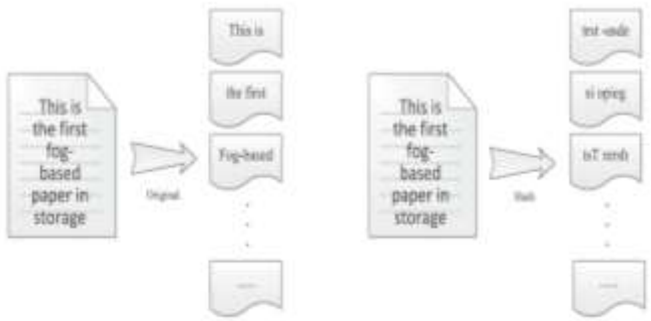

This is the first fog-based paper in storage
Original
This is
the first
Fog-based
This is the first fog-based paper in storage
Hash

Fig-6: Original transform vs. Hash Transform

In this the original code divides a sentence into different fragments it depends according to original sequence and the hash code is also divided the sentence into different fragments sequences. Here the hash Solomon code is used to improves the privacy protection and prevents the hackers from getting the information about the user's fragments.

2.5 Efficiency Analysis

The storage efficiency is a main part of index for storage of related algorithms. The systems with high storage efficiency can save more storage capacity. The storage industry networking association defines the storage efficiency as

$$StorageEfficiency = \frac{DataSpace}{DataSpace + CheckSpace}$$

Here the storage efficiency can be expressed as Es=K/K+M.

$$E_s = \frac{k}{k+m} = \frac{\frac{k}{m}}{\frac{k}{m}+1}$$

$$\lim_{\frac{k}{m}\to\infty} = \frac{\frac{k}{m}}{\frac{k}{m}+1} = 1$$

When the ratio of K and M increases the number of data blocks K also increases which includes the coding efficiency. The relationship between the K, M completes the equation the 2w>K+M. where w increases the consuming of the RAM also increases the reciprocal of w to the coding efficiency and it can be expressed as

$$E_c = \frac{\ln(k+m)}{\ln 2}$$

The comprehensive efficiency of the scheme can be expressed as

$$E_w = C_1\frac{\ln(k+m)}{\ln 2} + C_2\frac{k}{k+m}$$

Thus, the parameter of C1 and C2 are both related to the storage ratio. Setting the value of K corresponding to the summit for the whole efficiency of the scheme is the most suitable values.

3. EXPERIMENT AND RESULT

In this project the performance and feasibility of the three-layer server frame work based on the fog computing model through a different method including encoding, decoding and test of different sizes of data.

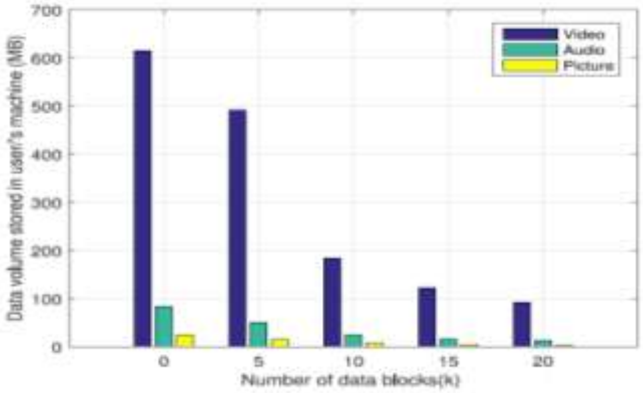


Chart-1: The local storage volume of different files.

Here we show the relationship between the number of blocks and data storage in users local machine using various kinds of data. Therefore, the parameter M represents the number of redundant data blocks and the parameter K represents the number of data blocks which is the need of the original data to be splitter. When the number of data blocks K increase the data stored in the user's local machine will be decreased. This means the more the number of data is the lesser the local storage.

Its performance differently when we use the different type of data the lager the volume of the data is the better effect of the experiment perform will be. Therefore, in the real scheme it is important to increase the value of the K to the user's storage. Before uploading the files, it must be merged as for the smaller files.

4. CONCLUSION

The cloud computing brings more benefits. Cloud storage which helps users to storage more numbers of data. Using the cloud storage users do not control the physical storage of their data it results in separation of the user's documents or data or information. In order prevent the problem of protecting the cloud storage; we have introduced the three-layer server frame work it is based on the fog computing model using the hash Solomon algorithm. The data can be stored in different blocks like cloud server, fog server, local machine and the code or data will be in the encode matrix. Cracking the encode matrix is impossible theoretically. So, the attacker cannot get the users information easily.

5. REFERENCES

[1] Tian Wang, Jiyuan Zhou, Xinlei Chen, Guojun Wang, Anfeng Liu, and Yang Liu, Member, IEEE, "A Three-Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing", IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, VOL. 2, NO. 1, FEBRUARY 2018.

[2] Y.Li,T.Wang,G.Wang,J.Liang,andH.Chen, "Efficient data collection in sensor-cloud system with multiple mobile sinks," in Proc. Adv. Serv. Comput., 10th Asia-Pac. Serv. Comput. Conf., 2016, pp. 130–143.

[3] L. Xiao, Q. Li, and J. Liu, "Survey on secure cloud storage," J. Data Acquis. Process., vol. 31, no. 3, pp. 464–472, 2016.

[4] J. Chase, R. Kaewpuang, W. Yonggang, and D. Niyato, "Joint virtual machine and bandwidth allocation in software defined network (sdn) and cloud computing environments," inProc.IEEEInt. Conf.Commun.,2014, pp. 2969–2974.

[5] H. Li, W. Sun, F. Li, and B. Wang, "Secure and privacy-preserving data storage service in public cloud," J. Comput. Res.Develop.,vol.51,no.7,pp.1397–1409,2014.