International Journal of Scientific Research and Engineering Development-– Volume 3 Issue 4, July-Aug 2020

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

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

Lakshmi Prasad[1], Ajeesh S[2],Smita C Thomas[3]

[1](P G Scholar, Department of Computer Science and Engineering,Mount Zion College of Engineering, Kadammanitta
Email: lakshmiprsd24@gmail.com)

[2](Assistant Professor, Department of CSE, Mount Zion College Of Engineering, Kadammanitta)

[3] (Research Scholar, Vels University,Chennai
Email: smitabejoy@gmail.com)

----------**********----------

Abstract:

Distributed computing and capacity gives clients to store and procedure their information in server farms. At the point when an association chooses to store information in cloud, it loses its entitlement to access to servers facilitating its data. Along these lines there is an opportunity of insider assaults. Existing security insurance systems are typically founded on encryption innovation, yet these sorts of techniques can't viably oppose assault from the cloud. To determine this issue, here proposes a three-layer stockpiling structure which depends on haze registering. The proposed engineering can both exploit distributed storage and secure the protection of information. In addition, Hash-Solomon code calculation is intended to partition information into sections. At that point, we can place a little piece of information in neighborhood machine and haze server so as to ensure the security and other in cloud. Additionally, in light of computational knowledge, this calculation can figure the dissemination extent put away in cloud, haze, and neighborhood machine, separately. By the hypothetical security investigation and trials, the possibility of our plan has been approved, which is actually a successful outcome to existing distributed storage plot.

Keywords **—Cloud computing, cloud storage, fog computing, privacy protection**

----------**********----------

I. INTRODUCTION

Cloud is only a gathering of servers and datacenters that are set at better places and these servers and datacenters are liable for giving on request administration to its clients with the assistance of web. The administration gave by cloud is absent on client's PC. Client needs to get to these administrations with assistance of web association through buying in them. The fundamental favorable position of Cloud figuring is that it takes out the requirement for client to be in same area where equipment, programming and extra room is truly present[2]. Cloud makes it

conceivable to store and access your information from anyplace whenever without agonizing over support. All the administrations are given to the client less expense. Client needs to pay as per the extra room utilized.

Distributed computing is a technique for conveying data innovation administrations in which assets are gotten from the Internet through online instruments and applications, rather than an immediate association with a server. It's called distributed computing in light of the fact that the data being gotten to is found in the cloud and doesn't require a client to be in a particular spot to access it. Distributed computing is turning out to be increasingly famous. It is utilized in numerous zones. So as to pull in most extreme clients, it needs to offer great quality types of assistance which ought to be secure and solid. Security turns into a major issue when any one stores its data to a stage which isn't straightforwardly constrained by client.

The utilization of cloud is straightforward, so everybody is moving information and application programming to cloud server farms. The Cloud Service Provider (CSP) ought to give security, uprightness, accessibility and privacy. In any case, CSP isn't giving. proper information administrations to client and put away information. The issues identified with distributed storage are information robbery, inaccessibility of information and information breaks. An information break happens when a programmer takes, uses, or discharges delicate data. Information is put away in the cloud shared by a huge number. The information is portable, that it very well may be moved starting with one area then onto the next. The cloud clients may not know about the information area. The private data is put away from the proprietor, it expands its powerlessness. The protection of cloud can't be ensured.

A noxious insider is a current or previous representative, contractual worker, or accomplice who has the privilege to get to organization information, and takes or harms the information. Protection depends on secure procedures, for example, solid access control, consistent checking forms and examines activities that lie beyond adequate capacities.

Conventional secure distributed storage answers for the above issues are normally concentrating on limitations to access or encryption of the information. These strategies can resolve a large portion of the issues. Be that as it may, the entirety of the arrangements can't tackle the assaults well, despite the fact that the calculation improves. Hence, here proposes a TLS conspire which depends on haze figuring model and structure a Hash-Solomon code dependent on Reed-Solomon code. Haze figuring is an all-inclusive processing model dependent on distributed computing which is made out of a great deal of hubs. These hubs have a specific stockpiling limit and preparing capacity.

In this plan, split client's information into three sections and each part is spared in the cloud server, the mist server and the client's neighborhood machine. Additionally, contingent upon the property of the Hash-Solomon code, the plan can guarantee the first information can't be recouped by halfway information. Utilizing Hash-Solomon code will deliver a piece of excess information squares which will be utilized in translating. Expanding the quantity of excess squares can build the unwavering quality of the capacity, however it likewise brings about extra information stockpiling. By fitting designation of the information, our plan can truly ensure the security of client's information. The Hash-

Solomon code needs complex estimation, which is done dependent on computational knowledge.

II. LITERATURE SURVEY

The significance part of security in distributed storage has pulled in a great deal of consideration regardless of for a solitary client or industry. There are a ton of explores about secure distributed storage models as of late. To understand the security issue in distributed computing, paper [1] proposed a protection saving and duplicate discouragement CBIR conspire utilizing encryption and watermarking strategies. This plan can secure the picture substance and picture highlights from the semi-legit cloud server, and deflect the picture client from illicitly appropriating the recovered pictures. Shen et al. think cloud is semi-trusted and propose a system for urban information sharing by misusing the quality based cryptography. The plan they proposed is make sure about and can oppose potential assaults [2]. Fu et al. propose a substance mindful inquiry plot, which can make semantic hunt more brilliant. The tests results show that their plan is productive [3].

In paper [4], Hou, Pu and Fan think about that in customary circumstance, client's information is put away through CSP, regardless of whether CSP is dependable, aggressors can even now get client's information on the off chance that they control the distributed storage the executives hub. To maintain a strategic distance from this issue, they propose an encoded file structure dependent on a deviated challenge-reaction validation instrument. At the point when client demands information from cloud server, the client sends a secret key to the server for recognizable proof. Thinking about it that the secret key might be captured, the structure utilizes unbalanced reaction mode. Hou, Wu, Zhen and Yang call attention to that the safe center of distributed storage is security and protection in conveyed framework. So they propose a safe virtual insurance plot dependent on SSL and Daoli in paper [5], [6]. By moving information over SSL and conveying Daoli on the cloud server, the framework encodes information before it is composed into the hard plate. In paper [7], Feng brings up that in paper [5], the weight of server will increment and information may spill during transmission in cloud servers.

Feng proposes a more concise scheme: encrypting data in closed cloud environment. Besides, it can achieve multi-point secure storage with one time encrypting. However, these encryption make search in cloud more difficult. Currently, searchable encryption is a hot topic in the field of cloud computing. Paper [8]–[10] give different solutions to this problem. Each of them achieves high accuracy, security and efficient.

III. EXISTING SYSTEM

In the existing model there is an authentication process for each user. The whole data is encrypted before storing to the cloud.

Authentication

User saves the data using username and password. By this unauthorized users cannot access the data. This has less security that the username and password can be hacked.

Encryption

In cryptography, encryption is the method by which plaintext or any other type of data is converted from a readable form to an encoded version that can be decoded using the key. It is one of the most important methods for providing security. The encryption process helps securing data even from providers.In the existing system XOR mechanism is used for encryption. This mechanism is very easy to hack because the length of the plain text and key size is same. So it is easy to find the key. Thus the encryption becomes useless.

IV. SECURE STORAGE USING FOG COMPUTING

So as to ensure client's security, we propose a TLS system dependent on haze registering model. The TSL system can give client a specific intensity of the executives and successfully ensure client's protection. As referenced, the inside assault is hard to stand up to. Conventional methodologies function admirably in understanding outside assault, however when CSP itself has issues, customary ways are for the most part invalid. Not quite the same as the conventional methodologies, in our plan, client's information is isolated into three distinctive size parts with encoding innovation. Every one of them will come up short on a piece of key data for secrecy. Consolidating with the mist processing model, the three pieces of information will be put away in the cloud server, the mist server and client's nearby machine as indicated by the request from huge to little.

By this method, the attacker cannot recover the user's original data even if he gets all the data from a certain server. As for the CSP, they also cannot get any useful information without the data stored in the fog server and local machine because both of the fog server and local machine are controlled by users.

As appeared in Figure the TLS structure utilizes mist server's stockpiling and information handling capacity. The engineering incorporates three layers, the cloud server, the haze server and the nearby machine. Every server spares a specific piece of information, the capacity extent is dictated by clients' distribution procedure. Right off the bat, client's information will be encoded on client's neighborhood machine. At that point, for instance, let 1%encoded information be put away in the machine. At that point transfer the rest of information to the mist server. Also, on the haze server, we do comparable activities to the information which originates from client's machine. There will be about 4% information put away in the mist server and afterward transfer the rest of to the cloud server. The above tasks depend on Hash-Solomon code. Hash-Solomon code is a sort of coding strategies dependent on Reed-Solomon code. In the wake of being encoded by Hash-Solomon code, the information will be separated into k parts and creates m repetitive information. Hash-Solomon code has such property, in these k+m parts of information, in the event that somebody has at any rate k parts, he can recoup the total information. In other word, it's not possible for anyone to recoup the total information with not as much as k parts of information. As indicated by this property of Hash-Solomon code, in our plan, we let close to k-1 pieces of information be put away in higher server which has bigger capacity limit and left the rest of put away in the lower server. Along these lines, the stealer can't recuperate the total information regardless of whether one of the three layers' information was taken. Accordingly we can guarantee the security of client's information.

Invalid Ratioisthe ratio of the number of failure data blocks to the number of data blocks which will be used in encoding. In other words, the ratio of the number of data blocks stored in lower server to the number of data blocks stored in the upper server. For example, the ratio of the number of data blocks stored in the local machine to the number of data blocks stored in the fog server. In the same way, the ratio of the number of data blocks stored in the fog server to the number of data blocks stored in the cloud server.

Maximal Invalid Ratioisthe maximal invalid ratio is the ratio of the number of invalid data to the number of all data blocks when the upper server can just recover the complete data by the data blocks stored in them. If there was one

more invalid data blocks, the upper server can't recover the complete data anymore.

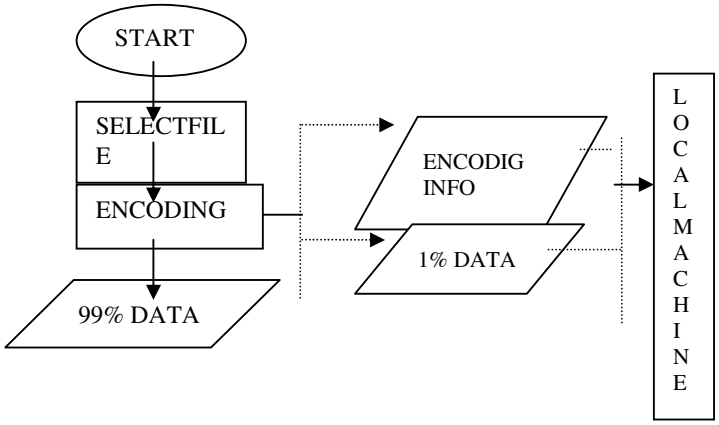
In Hash-Solomon code, the *Maximal Invalid Ratio* can be expressed as $m \div k+m$. For convenience, we just consider two layers situation. Assuming that there is x MB data which is prepared to save. After encoding, there will be $(k+m) \div m * x$ data. We prepare to save $r\%$ in the lower server. In order to avoid the upper server recovers the data, the value of k, m and r must satisfy the relationship:

$$m \div (k + m) \leq (k + m) \div k * r \quad (1)$$

Through functional transformation, the relationship between k m and r can be expressed as formula (2). We can see that if the parameter r is determined, the parameter k can be expressed by m. So we can only consider the ratio and the number of data blocks when we use our scheme.

$$k = (m - 2mr) + \sqrt{(2mr - m)^2 - 4m^2r^2}/2r \quad (2)$$

The parameter k is the number of blocks after data being divided, the parameter m is the number of redundant data blocks and the parameter r is the storage ratio of different servers. Besides, the fog server includes Computational Intelligence which can help the system with calculating the results of the values of k and m, because of the nodes in the fog server having its own computing power.



V. EFFICIENCY ANALYSIS

In past area, we have talked about the relationship of k and m. We find that the proportion of k and m is chosen once the capacity proportion is chosen. It implies that in the event that we set the capacity proportion as 20%, k = 3m. At that point we set k = 3, m = 1. In the genuine situation, information squares can't be put away halfway. In the above model, the lower server must store at any rate 2 squares, so the genuine stockpiling proportion is half, which is a long way from the 20%. So as to lessen mistake, we can leave k or m alone an enormous number. In any case, with the expanding of k, the encoding and disentangling proficiency will diminish, which will be demonstrated by tests in the following area. In this area, we will talk about how to adjust the capacity effectiveness and the coding proficiency. Finally, we propose an exhaustive record of the entire productivity of the plan.

The storage efficiency is an important index for a storage related algorithm. A good system with high storage efficiency can save storage capacity as much as possible. Storage Industry Networking Association defines the storage efficiency as:

$StorageEfficiency = DataSpace \div DataSpace + CheckSpace$

In the scheme, storage efficiency can be expressed as $Es = k \div k+m$. Then we can get the following formulas (4, 5).We can see that the storage efficiency will increase with the increment to the ratio of k and m. When the ratio of k and m increase, the number of data blocks (k) also increase, which influences the coding efficiency.

VI. ADVANTAGES AND DISADVANTAGES OF PROPOSED SYSTEM

The proposed system effectively protects user's privacy. The Fog server and local machine is controlled by user. It is worth noting that Hash-Solomon code has the following properties: in the $k+m$ data blocks, if we have at least k data blocks, we can recover the original data combining with the encoding matrix. But once the number of data blocks is less than k, it cannot be recovered. The Hash transform and encoding improves the privacy of the user.

Data loss can be occurred in local machine that includes 1% of data. Accessing of the data will be difficult from other locations. Encoding and decoding time is high for large data.

VII. CONCLUSION

The improvement of distributed computing presents to us a ton of advantages. Distributed storage is an advantageous innovation which causes clients to grow their capacity limit. Notwithstanding, distributed storage likewise causes a progression of secure issues. When utilizing distributed storage, clients don't really control the physical stockpiling of their information and it brings about the partition of possession and the board of information. So as to take care of the issue of security insurance in distributed storage, we propose a TLS system dependent on mist registering model and plan a Hash-Solomon calculation. Through the hypothetical security examination, the plan is end up being plausible. By apportioning the proportion of information squares put away in various servers sensibly, we can guarantee the security of information in every server. On another hand, splitting the encoding framework is outlandish hypothetically. In addition, utilizing hash change can secure the fragmentary data. Through the analysis test, this plan can productively finish encoding and interpreting without impact of the distributed storage effectiveness. Moreover, we plan a sensible complete effectiveness record, so as to accomplish the most extreme proficiency, and we additionally find that the Cauchy grid is increasingly productive in coding process.

REFERENCES

[1]. P. Mell and T. Grance, "The NIST definition of cloud computing," Nat.Inst. Stand. Technol., vol. 53, no. 6, pp. 50–50, 2009.

[2]. H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," Wireless Commun. Mobile Comput., vol. 13, no. 18, pp. 1587–1611, 2013.

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

[4]. 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.

[5]. Y. Li, T.Wang, G.Wang, J. Liang, and H. 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.

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

[7]. R. J. McEliece and D. V. Sarwate, "On sharing secrets and reed-solomon codes," Commun. ACM, vol. 24, no. 9, pp. 583–584, 1981.

