AN ADOPTIVE THREE LEVEL STOAGE FRAMEWORK FOR EFFICIENT IMPLEMENTATION OF CLOUD STORAGE IN FOG COMPUTING

Ms. V.kavitha [1], Mrs. V.Tejaswini[2*]

1 Final Year MCA Student, QIS College of Engineering and Technology, Ongole

2 Assistant Professor, MCA Dept., QIS College of Engineering and Technology, Ongole*

Abstract: Cloud computing emerges in recent technology for sharing information and in these days witness the improvement of cloud computing technology. With the dangerous development of unstructured information, cloud storage innovation shows signs of improvement advancement. In any case, in current stockpiling blueprint, client's information is completely put away in cloud servers. At the end of the day, clients lose their privilege of control on information and face protection spillage hazard. Conventional security assurance plans are typically founded on encryption innovation, yet these sorts of techniques can't successfully oppose assault from within cloud server. So as to take care of this issue, we propose a three-layer storage framework based on fog computing. The proposed structure can both exploit cloud storage and ensure the protection of information. Furthermore, Hash-Solomon code algorithm is intended to isolate information into various parts. At that point, we can put a little piece of information in neighborhood machine and mist server so as to secure the protection. Also, in view of computational insight, this calculation can register the appropriation extent put away in cloud, mist, and neighborhood machine, individually. Through the hypothetical wellbeing examination and exploratory assessment, the attainability of our plan has been approved, which is extremely an amazing enhancement to existing cloud storage conspire.

Keywords: *Cloud Computing, Hash-Solomon code algorithm, Correlation, Fog Computing.*

I. INTRODUCTION

Since the 21st century, computer technology has developed rapidly. Cloud computing, an emerging technology, was first proposed in SES 2006 (Search Engine Strategies 2006) by San Jose and defined by NIST (National Institute of Standards and Technology) [1]. Since it was proposed, cloud computing has attracted great attention from different sectors of society. Cloud computing has gradually matured through so many people's efforts [2]. Then there are some cloud-based technologies deriving from cloud computing. Cloud storage is an important part of them.

With the rapid development of network bandwidth, the volume of user's data is rising geometrically [3]. User's requirement cannot be satisfied by the capacity of local machine any more. Therefore, people try to find new methods to store their data. Pursuing more powerful storage capacity, a growing number of users select cloud storage. Storing data on a public cloud server is a trend in the future and the cloud storage technology will become widespread in a few years.

The privacy problem [4], is particularly significant among those security issues. In history, there were some famous cloud storage privacy leakage events. For example, Apples iCloud leakage event in 2014, numerous Hollywood actresses private photos stored in the clouds were stolen. This event caused an uproar [5], which was responsible for the users' anxiety directly. Subsequently, the Cloud Server Provider (CSP) will take place of user to manage the data. In consequence, user does not actually control the physical storage of their data, which results in the separation of ownership and management of data [6]. The CSP can freely access and search the data stored in the cloud. Meanwhile the attackers can also attack the CSP server to obtain the user's data. The above two cases both make users fell into the danger of information leakage and data loss. Traditional secure cloud storage solutions for the above problems are usually focusing on access restrictions or data encryption. These methods can actually eliminate most part of these problems. However, all of these solutions cannot solve the internal attack well, no matter how the algorithm improves. Therefore, we propose a TLS scheme based on fog computing model and design a Hash-Solomon code based on Reed-Solomon code [7], [8].

Fog computing is an extended computing model based on cloud computing which is composed of a lot of fog nodes [9]. These nodes have a certain storage capacity and processing capability [10].

In our scheme, we split user's data into three parts and separately save them in the cloud server, the fog server and the user's local machine. Besides, depending on the property [11], of the Hash-Solomon code, the scheme can ensure the original data cannot be recovered by partial data. On another hand, using Hash-Solomon code will produce a portion of redundant data blocks which will be used in decoding procedure. Increasing the number of redundant blocks can increase the reliability of the storage, but it also results in additional data storage [12,13]. By reasonable allocation of the data, our scheme can really protect the privacy of user' data.

The Hash-Solomon code needs complex calculation, which can be assisted with the Computational Intelligence (CI).

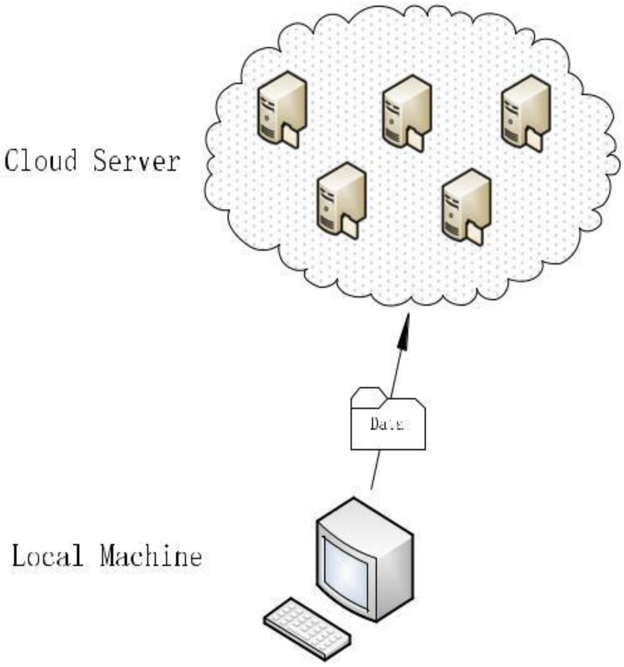


Fig: Traditional Cloud Storage Structure

Paradigms of CI have been successfully used in recent years to address various challenges, for example, the problems in Wireless sensor networks (WSNs) field. CI provides adaptive mechanisms that exhibit intelligent behavior in complex and dynamic environments likeWSNs [9]. Thus in our paper, we take advantage of CI to do some calculating works in the fog layer. Compared with traditional methods, our scheme can provide a higher privacy protection from interior, especially from the CSPs.

II RELATED WORK

SECURE CLOUD STORAGE BASED ON FOG COMPUTING

The security degree is an important metric to measure the quality of cloud storage system. Furthermore, data security is the most important part in cloud storage security and it includes three aspects: data privacy, data integrity and data availability. Ensuring data privacy and integrity has always been the focus of relevant researches [14]. On another hand, data privacy is also the most concerned part of the users. From a business perspective, company with high security degree will attract more users. Therefore improving security is an crucial goal no matter in academia or business. In this section, we will detailedly elaborate how the TLS framework protects the data privacy [15], the implementation details of work flow and the theoretical safety and efficiency analysis of the storage scheme.

A. Fog Computing

Our scheme is based on fog computing model, which an extension of cloud is computing. Fog computingwas firstly proposed by Ciscos Bonomi in 2011 [16]. In Bonomi's view, fog computing is similar to the cloud computing, the name of fog computing is very vivid. Compared to highly concentrated cloud computing, fog computing is closer to edge network and has many advantages as follows: broader geographical distributions, higher real-time and lowlatency [17]. In considering of these characters, fog computing is more suitable to the applications which are sensitive to delay. On another hand, compared to sensor nodes, fog computing nodes have a certain storage capacity and data processing capability [18][19], which can do some simple data processing, especially those applications based on geographical location. Thus we can deploy CI on the fog server to do some calculating works. Fog computing is usually a three-level architecture; the upmost is cloud computing layer which has powerful storage capacity and compute capability. The next level is fog computing layer. The fog computing layer serves as the middle layer of the fog computing model and plays a crucial role in transmission between cloud computing layer and sensor network layer. The fog nodes in fog computing layer have a certain storage capacity and compute capability. The bottom is wireless sensor network layer [20]. The main work of this layer is collecting data and uploading it to the fog server. Besides, the transfer rate between fog computing layer and other layers is faster than the rate directly between cloud layer and the bottom layer [21]–[23]. The introduction of fog computing can relief the cloud computing layer, improving the work efficiency. In our scheme, we take advantage of the fog computing model, adopt three-layer structure.

Furthermore, we replace the WSNs layer by user's local machine.

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

In order to protect user's privacy, we propose a TLS framework based on fog computing model. The TSL framework [22], can give user a certain power of management and effectively protect user's privacy. As mentioned, the interior attack is difficult to resist. Traditional approaches work well in solving outside attack, but when CSP itself has problems, traditional ways are all invalid. Different from the traditional approaches, in our scheme, user's data is divided into three different-size parts with encoding technology. Each of them will lack a part of key information for confidentiality [23]. Combining with the fog computing model, the three parts of data will be stored in the cloud server, the fog server and user's local machine according to the order from large to small. 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.

III. EXISTING SYSTEM

User uploads data to the cloud server directly. Subsequently, the Cloud Server Provider (CSP) will take place of user to manage the data. In consequence, users do not actually control the physical storage of their data, which results in the separation of ownership and management of data. In order to solve the privacy issue in cloud computing, previous researches proposed a privacy-preserving and copy-deterrence CBIR scheme using encryption and watermarking techniques. This scheme can protect the image content and image features well from the semi-honest cloud server, and deter the image user from illegally distributing the retrieved images. Previous works consider that in traditional situation, user's data is stored through CSP, even if CSP is trustworthy, attackers can still get user's data if they control the cloud storage management node. To avoid this problem, they propose an encrypted index structure based on an asymmetric challenge-response authentication mechanism. When user requests data from cloud server, the user sends a password to the server for identification. Taking it into consideration that the password may be intercepted, the structure uses asymmetric response mode.

Disadvantages:

The CSP can freely access and search the data stored in the cloud. Meanwhile the attackers can also attack the CSP server to obtain the user's data. The above two cases both make users fell into the danger of information leakage and data loss. Traditional secure cloud storage solutions for the above problems are usually focusing on access restrictions or data encryption.

IV. PROPOSED SYSTEM

However, all of these solutions cannot solve the internal attack well, no matter how the algorithm improves. There fore, we propose a TLS scheme based on fog computing model. Fog computing is an extended computing model based on cloud computing which is composed of a lot of fog nodes. These nodes have a certain storage capacity and processing capability. In our scheme, we split user's data into three parts and separately save them in the cloud server, the fog server and the user's local machine. We propose a new secure cloud storage scheme in this paper. By dividing file with specific code and combining with TLS framework based on fog computing model, we can achieve high degree privacy protection of data. It does not mean that we abandon the encryption technology. In our scheme encryption also help us to protect fine-grained secure of the data.

Advantages:

1. Compared with traditional methods, our scheme can provide a higher privacy protection from interior, especially from the CSPs.

2. From a business perspective, company with high security degree will attract more users.

3. Therefore improving security is an crucial goal no matter in academia or business. In this section, we will detailed elaborate how the TLS framework protects the data privacy, the implementation details of work flow and the theoretical safety and efficiency analysis of the storage scheme.

V ARCHITECTURE & SYSTEM COMPONENTS

Below architecture diagram represents mainly flow of request from the users to database through servers. In this scenario overall system is designed in three tiers separately using three layers called presentation layer, business layer, data link layer. This project was developed using 3-tier architecture.

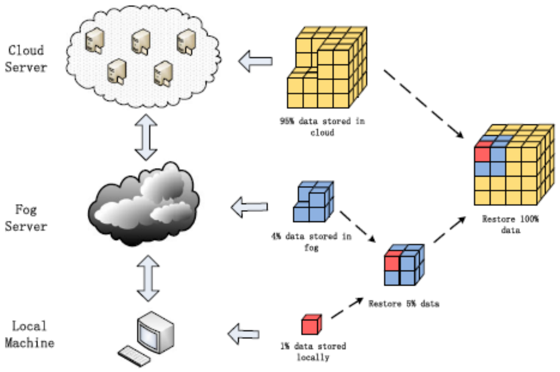


Fig: Three layer System Architecture

3-Tier Architecture:

The three-tier software architecture (a three layer architecture) emerged in the 1990s to overcome the limitations of the two-tier architecture. The third tier (middle tier server) is between the user interface (client) and the data management (server) components. This middle tier provides process management where business logic and rules are executed and can accommodate hundreds of users (as compared to only 100 users with the two tier architecture) by providing functions such as queuing, application execution, and database staging.

The three tier architecture is used when an effective distributed client/server design is needed that provides (when compared to the two tier) increased performance, flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user. These characteristics have made three layer architectures a popular choice for Internet applications and net-centric information systems.

DATA OWNER:

File owner will register with application and login with valid user name and password if verification is successful client can upload files to cloud server through fog server by keeping 1 percent of encrypted data at owner side and send 99 percent data to fog server for further processing.

Data owner will have permission to give key to user who wants to access data along with 1 percent data. In this process data owner will get information of any kind of activity happening to his data which is stored in cloud server.

FOGSERVER:

In this module fog server will act as small storage server and perform basic operations before sending data to cloud. In this second stage, after receiving the 99% data blocks from user's machine, these data blocks will be encoded again. These data blocks will be divided into smaller data blocks and generates new encoding information. Similarly, assuming that 4% data blocks and encoding information will be stored in the fog server. The remainder 95% data blocks will be uploaded to the cloud server.

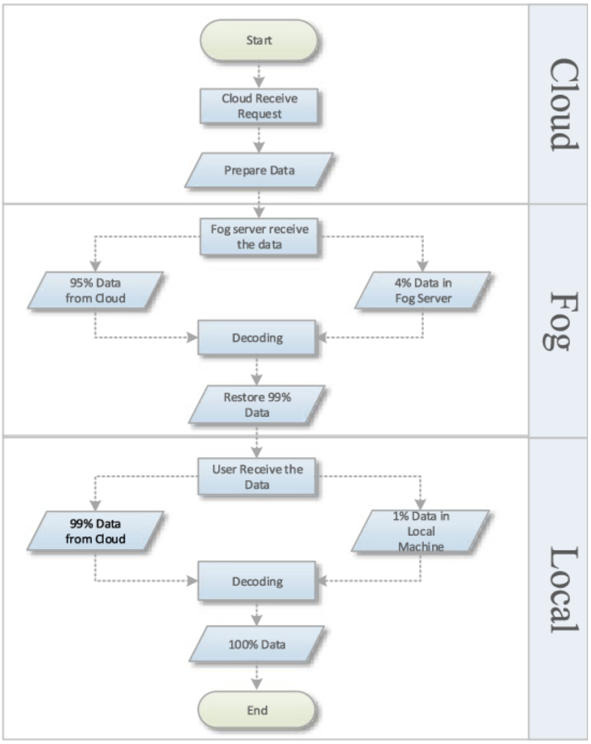
When user request for downloading data fog server will verify and send 4 percent of data to user.

CLOUD SERVER:

Cloud can login with valid user name and password the cloud storage server provides storage services to the registered clients for storing outsourced files. Storage server can view details of file uploaded by user which is received from fog server. In this process cloud server will only store 95 percent of data. When user requests for downloading data cloud server will store 95 percent of data.

Implementation Detail of Workflow

Stored Procedure*:* When user wants to store his file to the cloud server, the procedure is shown as Fig below. First of all, user's file will be encoded with Hash-Solomon code. And then, the file will be divided into several data blocks and the systems will also feedback encoding information simultaneously. Assuming that 1% data blocks and the encoding information will be stored locally. The remainder 99% data blocks will be uploaded to the fog server. Secondly, after receiving the 99% data blocks from user's machine, these data blocks will be encoded with Hash-Solomon again. These data blocks will be divided into smaller data blocks and generates new encoding information. Similarly, assuming that 4% data blocks and encoding information will be stored in the fog server. The remainder 95% data blocks will be uploaded to the cloud server. Thirdly, after cloud server received the data blocks form fog side; these data blocks will be distributed by cloud manage system [20]. Finally, the storage procedure ends when all the related information is recorded in different servers.



Download Procedure: When user wants to download his file from the cloud server, the procedure is shown in Fig. 4. Firstly, cloud server receives user's request and then integrates the data in different distributed servers. After integration, cloud server sends the 95% data to the fog server. Secondly, the fog server receives the data from the cloud server. Combining with the 4% data blocks of fog server and the encoding information, we can recover 99% data. Then the fog server returns the 99% data to the user. Thirdly, the user receives the data from fog server. User can get the complete data by repeating the above steps.

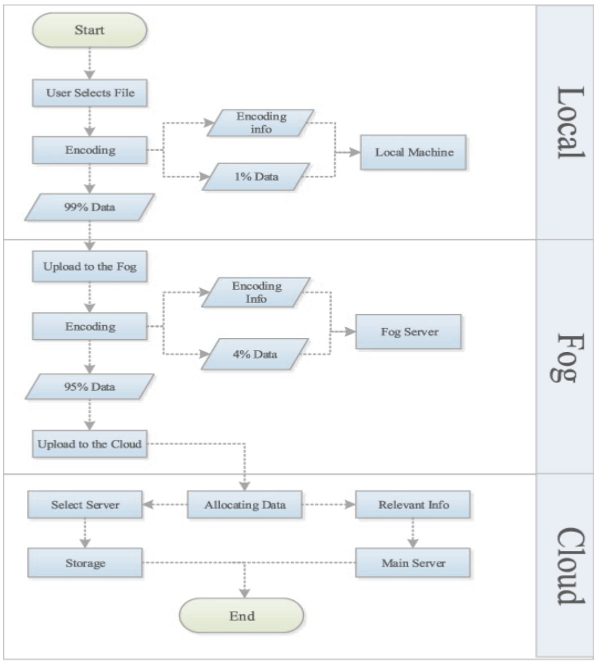


Fig: Download Procedure

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:

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

In our scheme, storage efficiency can be expressed as E_s = $k/k+m$. 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. we know that when the ratio of k and m increase, the number of data blocks (k) also increase, which influences the coding efficiency.

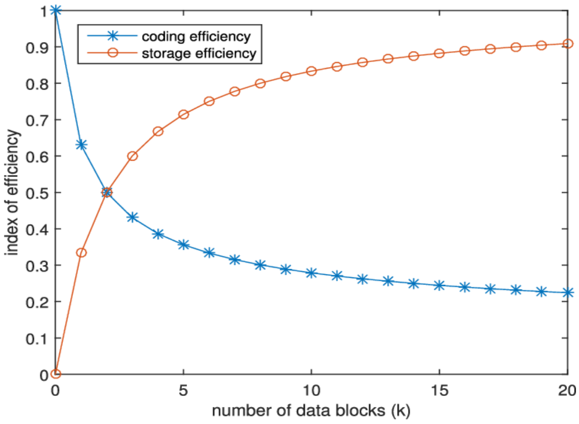


Fig: Diagram of the influence of the number of data blocks (k) to the efficiency of storage and coding.

VI. EXPERIMENT AND ANALYSIS

In this section, we evaluate the performance and feasibility of the TLS framework based on fog computing model through a series of tests, including encoding, decoding and test of different sizes of data.

A. Experimental Environment

All of the experiments in this paper were conducted by simulation and the environmental parameters are shown as Table II. There are three types files which are listed as flows: picture (.NEF, 24.3 MB), audio (.MP3, 84.2 MB) and video (.RMVB, 615 MB).

All the experiments in this paper use ’one more block’ principle which means the lower server only saving $m + 1$ data blocks. In this way, the scheme can ensure the privacy of data and reduce the storage pressure of the lower servers at the same time.

B. Experiment Results

Fig. 10 shows the relationship between data storage in user’s machine and the number of blocks while using different kinds of data. The parameter m represents the number of redundant data blocks while the parameter k represents the number of data blocks which we want the original data be divided into. Note that the value of m is set as 2 in this part. As we can see, when

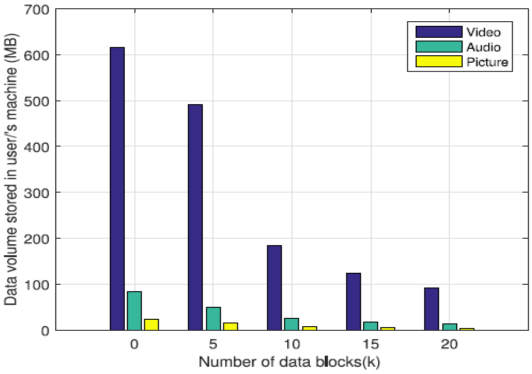


Fig. 10. The local storage volume of different files.

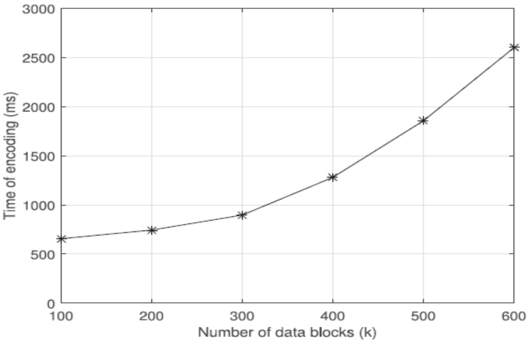


Fig. 11. Relationship between time of encoding and the number of k.

The number of data blocks *k* increases, the data volume stored in user's local machine decreases. It means that the more the number of data blocks is, the smaller the local storage pressure is. On another hand, our method performs differently when using different volume of data. The larger the volume of the data is, the better effect our method performs in the experiment. Therefore, in the real scenario, it is of vital importance to increase the value of *k* to alleviate user's storage pressure. As for small files, merging files before uploading is necessary.

Fig. 11 shows the tendency of encoding time with different number of data blocks. The value of *m* is also set as 2. When the number of data blocks *k* increases, the encoding time grows exponentially. Accordingly, in the real scenario, we should consider delay degree that user can endure and adjust the value of *k* according to the user's machine performance dynamically. The relationship between decoding time and number of data blocks is shown in Fig. 12. Both the value of *m* and the value of removed data is set as 2. When the number of data blocks *k* increases from 100 to 600, the decoding time increases at express speed. As we can see, the decoding process costs more time than the encoding process does, so we should pay more attention to enhance decoding efficiency in real scenario.

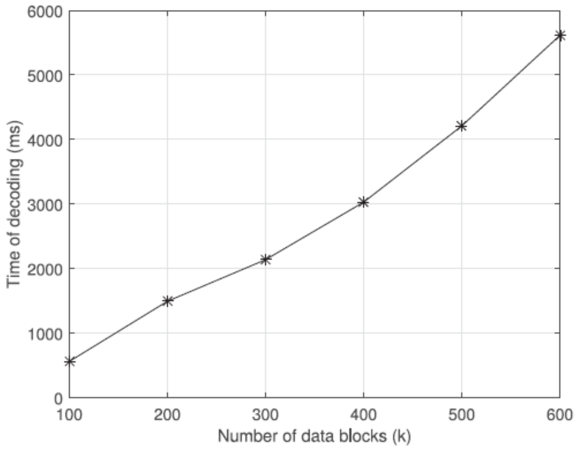


Fig. 12. Relationship between time of decoding and the number of k.

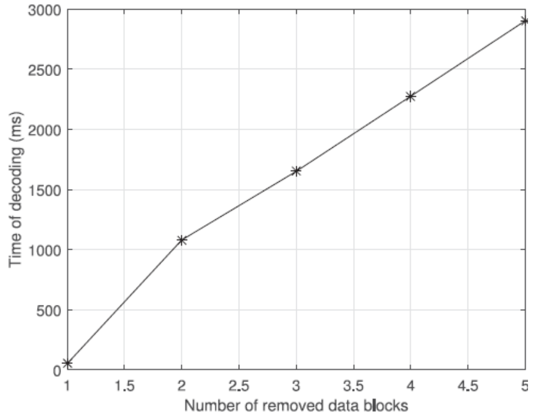


Fig. 13. Relationship between time of decoding and the number of removed data.

In the Fig. 13, we present the tendency of decoding time with different number of removed data from 1 to 5. The value of *k* is set as 100 and the value of *m* is set as 5. In the real scenario, the ratio of *m* and *k* should be very small to relieve the user's storage pressure. What's more, the number of removed data should be smaller than *m*, otherwise, system will be errorreporting. On another hand, the decoding time increases with the increment of the number of removed data, which means that we should download all of the data from the upper server as much as possible to maximize the decoding efficiency.

The Hash-Solomon code is the key to the whole efficiency of our scheme. Therefore, find a better coding matrix is of vital

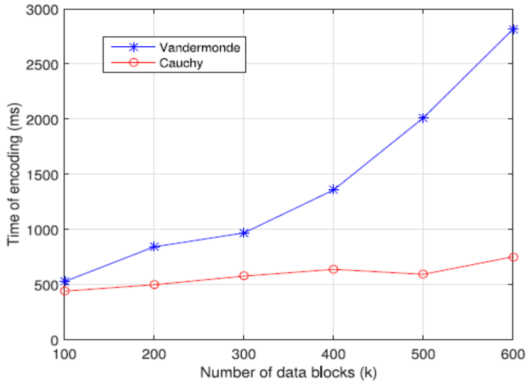


Fig. 14. Cauchy matrix vs. Vandermonde matrix.

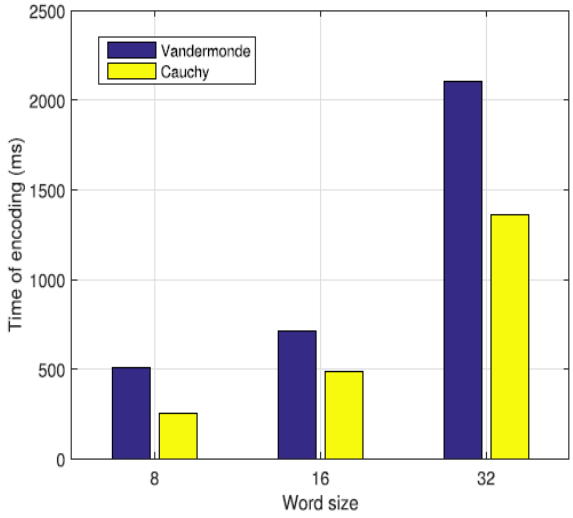


Fig. 15. Relationship between time of encoding and word size of Galois field.

importance. The code matrix can be chosen from Vandermonde matrix and Cauchy matrix. Different fromVandermonde matrix, Cauchy matrix uses AND operation and XOR logical operation. In Cauchy's way, coding efficiency improves. Besides, the complexity decrease from $O(n3)$ to $O(n2)$. As shown in the Fig. 14, we present the two tendencies of encoding time with different number of data blocks k from 100 to 600. The value of m is set as 2. We can see that the encoding time raises with the increase of the number of data blocks k, no matter Vandermonde or Cauchy. On another hand, the Cauchy matrix has better performance than Vandermonde matrix. The time cost of Cauchy always less than the Vandermonde. When the number of k is very large, the cost of Vandermonde raises sharply while the cost of Cauchy increases slightly.

In the Section III, the coding efficiency is related to the ω in Galois field $GF(2\omega)$. As shown in Fig. 15, we present the encoding time with different values of ω. Besides, we also consider the comparison of Vandermonde and Cauchy. As shown in the Fig. 15, we set the value of ω as 8, 16 and 32. As we can see, no matter Vandermonde or Cauchy, the cost of encoding time increases with the increase of ω.

VII RESULT

The development of cloud computing brings us a lot of benefits. Cloud storage is a convenient technology which helps users to expand their′ storage capacity. However, cloud storage also causes a series of secure problems. When using cloud storage, users do not actually control the physical storage′ of their data and it results in the separation of ownership and management of data. In order to solve the problem of privacy protection in cloud storage, we′ propose a TLS framework based on fog computing model and design a Hash-Solomon algorithm. Through the theoretical safety analysis, the scheme is proved to be feasible. ′ CONTINUE By allocating the ratio of data blocks stored in different servers′ reasonably, we can ensure the privacy of data in each server. On another hand, cracking the encoding matrix is impossible theoretically

VIII. CONCLUSION

The improvement of cloud computing presents to us a great deal of advantages. Cloud storage is an advantageous innovation which helps clients to extend their capacity limit. Nonetheless, cloud storage too causes a progression of secure issues. When utilizing distributed storage, clients don't really control the physical stockpiling of their information what's more; it results in the partition of proprietorship and the board of information. So as to take care of the issue of security insurance in cloud capacity, we propose a TLS system dependent on mist registering model and structure a Hash-Solomon calculation. Through the hypothetical wellbeing investigation, the plan is turned out to be attainable. By allotting the proportion of information squares put away in various servers sensibly, we can guarantee the protection of information in every server.

On another hand, breaking the encoding grid is outlandish hypothetically. Additionally, utilizing hash change can secure the fragmentary data. Through the analysis test, this plan can productively total encoding and deciphering without impact of the distributed storage proficiency. Moreover, we plan a sensible far reaching effectiveness list, all together to accomplish the most extreme proficiency, and we additionally find that the

Cauchy grid is increasingly effective in coding process.

IX. 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.

[8] J. S. Plank, "T1: Erasure codes for storage applications," in Proc. 4th USENIX Conf. File Storage Technol., 2005, pp. 1–74.

[9] R. Kulkarni, A. Forster, and G. Venayagamoorthy,"Computational intelligence in wireless sensor networks: A survey," IEEE Commun. Surv. Tuts., vol. 13, no. 1, pp. 68–96, First Quarter 2011.

[10] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, “A privacypreserving and copy-deterrence content-based image retrieval scheme in cloud computing,” IEEE Trans. Inf. Forensics Security, vol. 11, no. 11, pp. 2594–2608, Nov. 2016.

[11] J. Shen, D. Liu, J. Shen, Q. Liu, and X. Sun, “A secure cloud-assisted urban data sharing framework for ubiquitous-cities,” Pervasive Mobile Comput., vol. 41, pp. 219–230, 2017.

[12] Z. Fu, F. Huang, K. Ren, J.Weng, and C.Wang, “Privacy-preserving smart semantic search based on conceptual graphs over encrypted outsourced data,” IEEE Trans. Inf. Forensics Security, vol. 12, no. 8, pp. 1874–1884, Aug. 2017.

[13] J. Hou, C. Piao, and T. Fan, “Privacy preservation cloud storage architecture research,” J. Hebei Acad. Sci., vol. 30, no. 2, pp. 45–48, 2013.

[14] Q. Hou, Y. Wu, W. Zheng, and G. Yang, “A method on protection of user data privacy in cloud storage platform,” J. Comput. Res. Develop., vol. 48, no. 7, pp. 1146–1154, 2011.

[15] P. Barham et al., “Xen and the art of virtualization,” ACM SIGOPS Oper.Syst. Rev., vol. 37, no. 5, pp. 164–177, 2003.

[16] G. Feng, “A data privacy protection scheme of cloud storage,” vol. 14, no. 12, pp. 174–176, 2015.

[17] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, “Toward efficient multi keyword fuzzy search over encrypted outsourced data with accuracy improvement,” IEEE Trans. Inf. Forensics Security, vol. 11, no. 12, pp. 2706–2716, Dec. 2016.

[18] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, “Enabling personalized search over encrypted outsourced data with efficiency improvement,” IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 9, pp. 2546–2559, Sep. 2016.

[19] Z. Xia, X. Wang, X. Sun, and Q. Wang, “A secure and dynamic multikeyword ranked search scheme over encrypted cloud data,” IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 2, pp. 340–352, Feb. 2016.

[20] Z. Fu, F. Huang, X. Sun, A. Vasilakos, and C.-N. Yang, “Enabling semantic search based on conceptual graphs over encrypted outsourced data,” IEEE Trans. Serv. Comput.. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TSC.2016.2622697

[21] G. Kulkarni, R.Waghmare, R. Palwe, V.Waykule, H. Bankar, and K. Koli, “Cloud storage architecture,” in Proc. 7th Int. Conf. Telecommun. Syst., Serv., Appl., 2012, pp. 76–81.

[22] C.Wang, S. S. Chow, Q.Wang, K. Ren, andW. Lou, “Privacy-preserving public auditing for secure cloud storage,” IEEE Trans. Comput., vol. 62, no. 2, pp. 362–375, Feb. 2013.

[23] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, “An efficient public auditing protocol with novel dynamic structure for cloud data,” IEEE Trans. Inf. Forensics Security, vol. 12, no. 10, pp. 2402–2415, Oct. 2017.

Authors Profile

Ms. **V.kavitha** pursuing MCA 3rd year in Qis College and Engineering and Technology in Department of Master of Computer Applications, Ongole.

Mrs. **V.Tejaswini** is currently working as an Assistant Professor in Department of Master of Computer Applications in QIS College of Engineering & Technology.