DLINE JOURNALS

Journal of Networking Technology

Print ISSN: 0976-898X
Online ISSN: 0976-8998

JNT 2024; 15 (1)
https://doi.org/10.6025/jnt/2024/15/1/11-21

A Secure Privacy-preserving Three-layer Framework for Cloud Storage Based on Fog Computing

Kousalya, A
Sri Krishna College of Engineering and Technology
Coimbatore, India
kousalyaa@skcet.ac.in

Karpagavadivu, K
Dr. N.G.P Institute of Technology
Coimbatore, India
karpagavadivu@drngpit.ac.in

Received: 3 September 2023
Revised: 29 November 2023
Accepted: 11 December 2023

ABSTRACT

Witnessing the rapid growth in technology development in recent years, there is a need to store the huge amounts of data generated and stored. Cloud storage is widely used for storing large, structured, unstructured data. Cloud computing services are provided by third-party authorities to store all the data in a distributed framework. Data stored in the cloud has to be protected to guarantee the security of its clients. However, Cloud Service Providers (CSP) control the data stored by the users, thus separating ownership and management of data. The current schema uses traditional encryption technology, which is not effective in handling the user's data without privacy leakage. Hence, we propose a three-layer framework based on fog computing, an extension of cloud computing that carries out a substantial amount of computation, storage, and communication locally and routed over the internet. In this framework, we implement the Hash-Solomon code algorithm to isolate the data into several parts stored in a local machine, fog server and cloud server. This can ensure security from internal attackers. Moreover, Computational Intelligence computes the distribution proportion in each server. The plausibility has been validated through experimental analysis, a ground-breaking supplement to existing distributed storage conspire.

Keywords: *Cloud Computing, Security, Fog Computing*

1. Introduction

Since the 21st century, technology has developed rapidly. Cloud computing provides a great advantage and gains great attention from different sectors of society. Cloud computing has gradually matured through numerous people's efforts. There

are various cloud-based technologies deriving from cloud computing. Cloud storage is a crucial area of them. With the rapid development of network bandwidth, the amount of user data is rising geometrically.

The local machine's capacity can no longer satisfy users' requirements. Therefore, people plan to find new methods to store their data. Many users switch to cloud storage because of the more powerful storage capacity. Storing data on a public cloud server is also a trend in the long run, and so cloud storage technology will become widespread in some years. Cloud storage could also be a cloud computing system that provides data storage and management services. With a cluster of applications, network technology, and distributed system technology, cloud storage makes several varied storage devices work together. However, cloud storage services still feature tons of security problems. In security issues, the privacy problem is particularly significant. In history, there have been some famous cloud storage privacy leakage events. For example, in Apple's iCloud leakage event in 2014, numerous Hollywood actresses' private photos stored within the clouds were stolen. This event caused uproar, which was responsible for the users' anxiety about the privacy of their data stored within the cloud server.

The Cloud Server Provider (CSP) will take the user's place to manage the data. Consequently, users don't control the physical storage of their data, which leads to the separation of ownership and management of knowledge. The CSP can freely access and search the knowledge stored within the cloud. Meanwhile, the attackers can even attack the CSP server to steal the user's data. The above two cases both make users fall into the danger of knowledge leakage and data loss. Traditional secure cloud storage solutions for the above problems are usually specializing in access restrictions or encryption. These methods can eliminate most parts of these problems. However, these solutions won't provide solutions for internal attackers, even if we improvise the algorithm. Therefore, we propose a TLS scheme-supported fog computing model and style a Hash-Solomon code-supported Reed-Solomon code. Fog computing is an extended computing model supported by cloud computing, which consists of many fog nodes. These nodes have a particular storage capacity and processing capability. In our scheme, we split the user's data into three parts and separately save them within the cloud server, the fog server and, thus, the user's local machine. Besides, depending on the property of the Hash-Solomon code, the scheme can confirm partial data cannot recover the first data. On the other hand, using the Hash-Solomon code will produce several redundant data blocks, which may be utilized in the decoding procedure. Increasing the quantity of redundant blocks can increase the reliability of the storage, but it also results in additional data storage. The Hash-Solomon code algorithm needs complex calculations and can be assisted with Computational Intelligence. Paradigms of CI have been successfully utilized in recent years to address various challenges, for example, the problems in the wireless sensor networks (WSNs) field. CI provides adaptive mechanisms that exhibit intelligent behavior in complex, dynamic environments like WSNs. Thus, we make the foremost of CI to undertake and do some calculating works within the fog layer. Compared with traditional methods, our scheme can provide far better privacy protection from the interior, especially from the CSPs.

2. Literature Survey

Information deduplication is a key strategy to improve capacity productivity in distributed computing. By directing repetitive records toward a solitary duplicate, cloud specialist co-ops incredibly lessen their extra room just as information move costs. [1]. Even though the traditional deduplication approach has been adopted widely, it comes with a high risk of losing data confidentiality because of the data storage models in cloud computing. The proposed TEE (Trusted Execution Environment) based secure deduplication scheme is used to secure cloud storage. Each cloud client is allotted a benefit set; the deduplication can be performed if the cloud clients have the right benefit. Moreover, the scheme augments the convergent encryption with the user's privileges. It relies on TEE to provide secure key management, which improves the ability of such cryptosystems to resist chosen plaintext attacks and ciphertext attacks. A security probe indicates that the scheme is secure enough to support data deduplication and protect sensitive data confidentiality. Moreover, we execute a model of the plan and assess the exhibition of the TEE model; tests show that the overhead of our plan is viable in practical conditions.

Recent years have witnessed the development of cloud computing technology. With the explosive growth of unstructured data, cloud storage technology is getting more attention and developing

better. However, the user's data is stored in cloud servers in the current storage schema. In other words, users lose their right to control over data and face privacy leakage risks. Traditional privacy protection schemes are usually based on encryption technology, but these methods cannot effectively resist attacks from inside a cloud server. To solve this problem, we propose a three-layer storage framework based on fog computing.[2] The proposed framework can fully utilise cloud storage and protect data privacy. Besides, the Hash-Solomon code algorithm divides data into different parts. Then, we can put a small part of the data in a local machine and fog server to protect privacy. Moreover, based on computational intelligence, this algorithm can compute the distribution proportion stored in cloud, fog, and local machines. Through theoretical safety analysis and experimental evaluation, the feasibility of our scheme has been validated, which is a powerful supplement to the existing cloud storage scheme.

With the increasing importance of images in people's daily lives, content-based image retrieval (CBIR) has been widely studied [3, 4]. Compared with text documents, images consume much more storage space. Hence, its maintenance is considered a typical example of cloud storage outsourcing. For privacy-preserving purposes, sensitive images, such as medical and personal images, must be encrypted before outsourcing, making the CBIR technologies in the plaintext domain unusable. This paper proposed a scheme that supports CBIR over encrypted images without leaking sensitive information to the cloud server. First, feature vectors are extracted to represent the corresponding images. After that, the pre-filter tables are constructed by locality-sensitive hashing to increase search efficiency. Moreover, the secure kNN *algorithm protects the feature vectors, and image pixels are encrypted by a standard stream cipher. In addition, considering that the authorized query users may illegally copy and distribute the retrieved images to someone unauthorized, we propose a watermark-based protocol to deter such illegal distributions. In our watermark-based protocol, a unique watermark is directly embedded into the encrypted images by the cloud server before images are sent to the querying user. Hence, when an image copy is found, the unlawful query user who distributed the image can be traced by the watermark extraction. The security analysis and the experiments show the security and efficiency of the proposed scheme.*

Wireless sensor networks (WSNs) are networks of distributed autonomous devices that can sense or monitor physical or environmental conditions cooperatively [5]. WSNs face many challenges, mainly caused by communication failures, storage and computational constraints and limited power supply. In recent years, paradigms of computational intelligence (CI) have been successfully used to address various challenges, such as data aggregation and fusion, energy-aware routing, task scheduling, security, optimal deployment and localization. CI provides adaptive mechanisms that exhibit intelligent behavior in complex, dynamic environments like WSNs. CI brings about flexibility, autonomous behavior, and robustness against topology changes, communication failures and scenario changes. However, WSN developers are usually not or not completely aware of the potential CI algorithms offer [6]. On the other side, CI researchers are not familiar with all the real problems and subtle requirements of WSNs. This mismatch makes collaboration and development difficult. This paper intends to close this gap and foster collaboration by offering a detailed introduction to WSNs and their properties. An extensive survey of CI applications to various problems in WSNs from multiple research areas and publication venues is presented in the paper. Besides, a discussion on the advantages and disadvantages of CI algorithms over traditional WSN solutions is offered. In addition, a general evaluation of CI algorithms is presented, which will serve as a guide for using CI algorithms for WSNs.

Cloud computing extends the data processing and storage ability of wireless sensor networks (WSNs). However, due to the weak communication ability of WSNs, how to upload the sensed data to the Cloud within a limited time becomes a bottleneck of sensor-cloud systems. To solve this problem, we propose using multiple mobile sinks to help upload data from WSNs to the Cloud. An efficient algorithm is designed to schedule the multiple mobile sinks with several provable properties. We conduct extensive simulations to evaluate the performance of the proposed algorithm. The results show that our algorithm can upload the data from WSNs to the Cloud within the limited latency and minimize energy consumption.

Shamir's scheme for sharing secrets is closely related to Reed-Solomon coding schemes. Decoding algorithms for Reed-Solomon codes provide extensions and generalizations of Shamir's method.

Cloud computing has been gradually considered the most significant turning point in the development of information technology during the past few years. People reap the benefits from the cloud, such as ubiquitous and flexible access, considerable capital expenditure savings, pay-as-you-go computing resources configuration, etc. Many companies, organizations, and individual users have adopted the public cloud storage service to facilitate their business operations, research, or everyday needs. However, in the outsourcing cloud computing model, users' physical control of the underlying infrastructure, including the system hardware and lower levels of the software stack, is shifted to third-party, public cloud service providers; in addition, the sensitive data of users are also outsourced to and stored in the cloud; thus, the potential private information leakage and integrity of the outsourced data is one of the primary concerns for the cloud users. To build users' confidence in such a cloud storage service paradigm, tons of attention has been drawn, and several related problems have been studied extensively in the literature, such as fine-grained cloud data access control mechanisms, secure search over encrypted cloud data, outsourced data integrity auditing, secure deletion for cloud data, etc., which ensures that cloud users enjoy the convenience the cloud offers in a privacy-preserving way. Otherwise, the cloud will become merely remote storage that provides limited value to all parties [7]. This paper focuses on the enabling and critical cloud computing security protection techniques and surveys on the recent research in these areas. In addition, we further point out some unsolved but important challenging issues and hopefully provide insight into their possible solutions. [8]

Security and privacy are critical for cloud storage because extensive privacy information, such as personal data, is involved in services. Important progress has recently been focused on secure cloud storage, including data deduplication, oblivious storage, data encryption and ciphertext searching, and data integrity audit [9,10,11]. First, secure data deduplication is discussed for cloud storage to address multiple types of attacks, such as identifying file attacks, which reduces the overhead to cloud servers and benefits end-users. Recent progress has also been investigated on oblivious storage, which introduces the partition-based framework and asynchronicity to reduce the computation overhead, providing privacy protection and scalability for cloud storage [12]. Then, inspired by traditional encryption techniques, cloud encryption and ciphertext searching are considered, such as the ciphertext-policy attribute-based encryption that provides search services for encrypted data and avoids privacy leakage. With key-exposure resistance, a data integrity audit system provides security, efficiency and verifiability for cloud storage. In summary, the advanced research of the above security techniques is discussed, and future research on improving the security and privacy of commercial cloud storage systems is explored.

3. Existing System

In the existing system, the data is encrypted and encoded using Reed-Solomon code, which is notable for detecting and fixing several symbol errors. The algorithm encodes data and stores data in the cloud. Hence, it leads to internal attacks with the help of hacking the secret key. Also, the data owner or the cloud server does not authorise data.

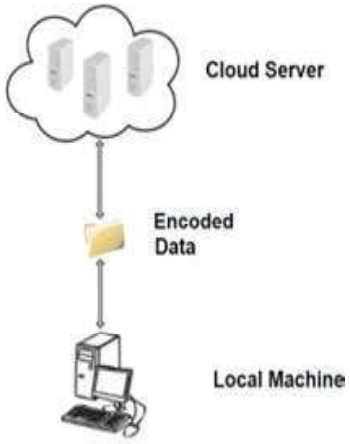

Figure 1. Existing System

Drawbacks of the existing system

• Unlike BCH codes, RS codes in BPSK modulation schemes perform poorly. The Reed-Solomon Codes Bit Error Ratio (BER) is weaker than the BCH codes.

• The data is unsafe due to zero authorization.

• Hashing is not implemented, leading to internal hacks with the secret key through hacking.

• Cloud Service Providers have full access to the user's data Hence, there is a separation of ownership and management of data.

4. Proposed System

The proposed system is a three-layer architecture for privacy preservation and security in the cloud using a fog server and computational Intelligence. The anticipated structure can exploit distributed storage and secure information protection.

Moreover, Hash-Solomon code calculation separates the information into various parts. At that point, we can place a little piece of information in a nearby machine and fog server to secure the protection.

Additionally, given computational insight, this calculation can figure the dispersion extent put away in a cloud and nearby machines separately.

System Architecture

A three-layer secure framework based on a fog computing model is designed to protect users' privacy. The TLS framework will effectively protect users' privacy and give certain management power to the users. As stated earlier, the internal attack is difficult to resist. Traditional approaches work well in solving outside attacks, but when CSP has problems, traditional ways are invalid. In our system, the user's data are divided into three different-sized parts with encoding technology, which differs from conventional approaches. Each of these parts will lack a part of key information for confidentiality purposes. Combining with the fog computing model, the three parts of data will be stored in the cloud server, the fog server and the 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 particular server. Because of this method of storing data in different parts, cloud service providers cannot get useful information without the data in the local machine and fog server since the users control both.

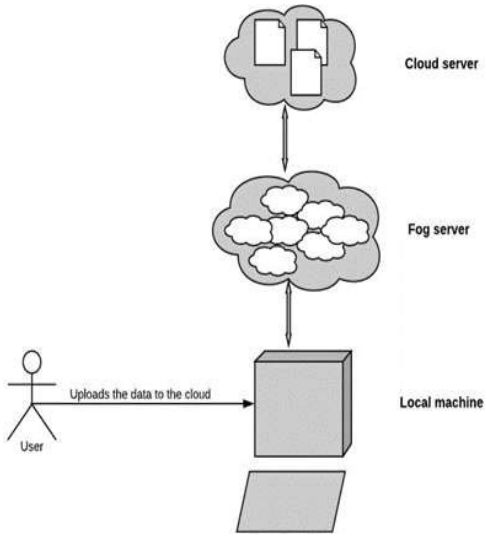

Figure 2 .Proposed Three-layer framework

The TLS framework fully uses the fog server's storage and data processing capability. The architecture consists of three layers: the cloud server, the fog server and the local machine. Each of these servers stores a certain part of data, and this storage proportion is determined by user allocation strategy.

Module description

The process of preserving privacy in cloud storage is done through various modules. The modules involved are as follows:

(i) Implementation Details of workflow

- Storage Procedure Module

(ii) Download Procedure Module Algorithm implementation module

Storage procedure module

In this module, the user stores the file to the cloud server. Initially, the user file will be encoded using the Hash-Solomon code. Then, the file will be divided into many data blocks, and simulta

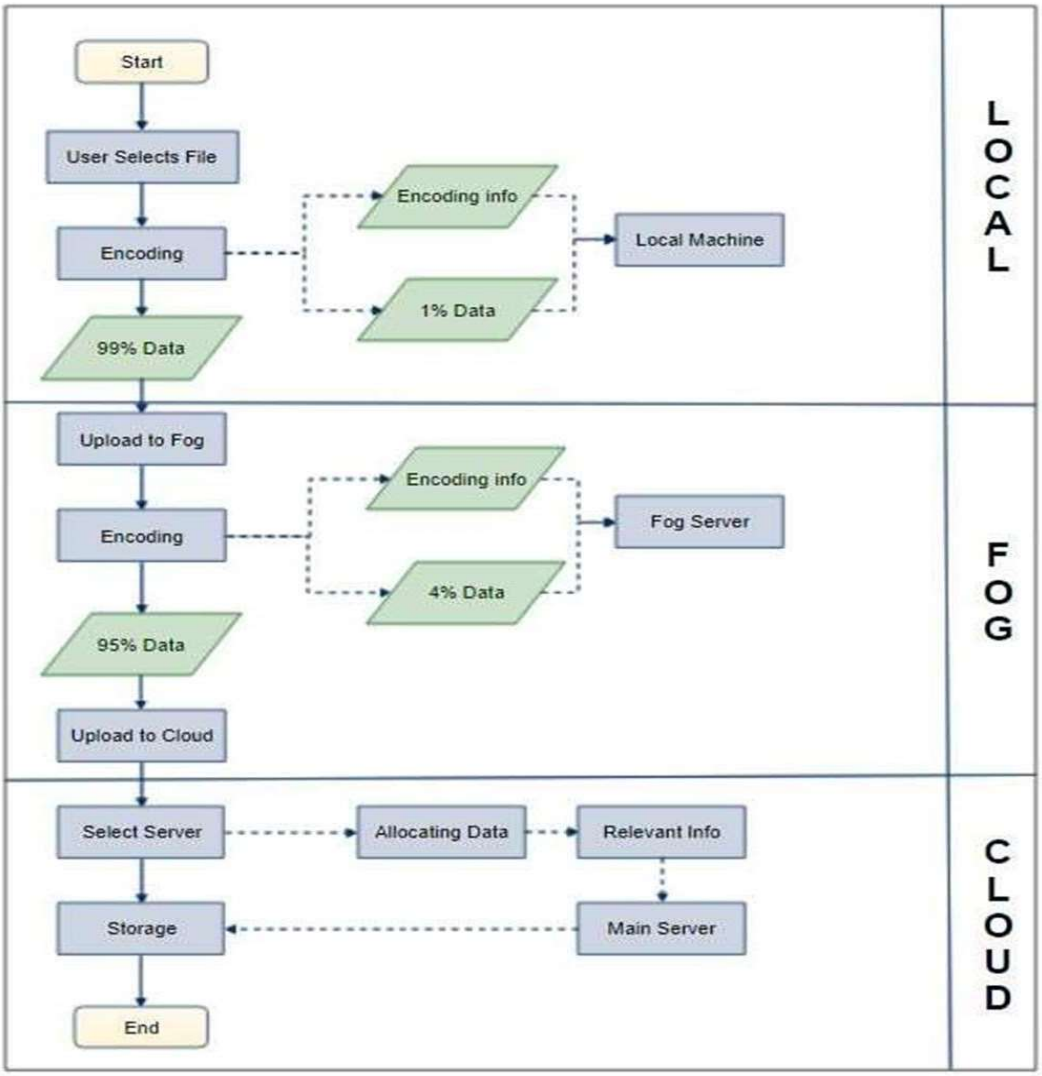

Figure 3. Storage process

neously, the system will provide feedback on the encoded information. They assume that a minimum amount of data blocks and the encoding information will be stored locally. The remaining huge number of data blocks will be uploaded to the fog server after receiving the huge data blocks from the user's machine. By using the Hash-Solomon code, the data blocks will be encoded again. These data blocks will be divided into smaller ones, generating new encoding information. They assumed that the minimum level of data blocks and encoding information, which is larger than the local machine, would be stored in the fog server. The remaining large data blocks will be uploaded to the cloud server. Finally, after the cloud server receives the data blocks from the fog server, these data blocks will be distributed by using a cloud management system. The storage procedure ends when all the related information is recorded on different servers.

Download procedure module

Suppose a user wants to retrieve his/her data from the cloud server. The user first sends the request to the cloud server. The cloud server receives the user request and combines the data in the distributed servers. After combining, the cloud server sends a large amount of data to the fog server. then the fog server receives the data from the cloud server. Combined with the small data blocks of the fog server and the encoding information, we can recover an extensive data set from both cloud and fog. Finally, the user receives the complete data by combining the large data set with the minimum data available in the local machine. Users can get the complete data by repeating the above steps.

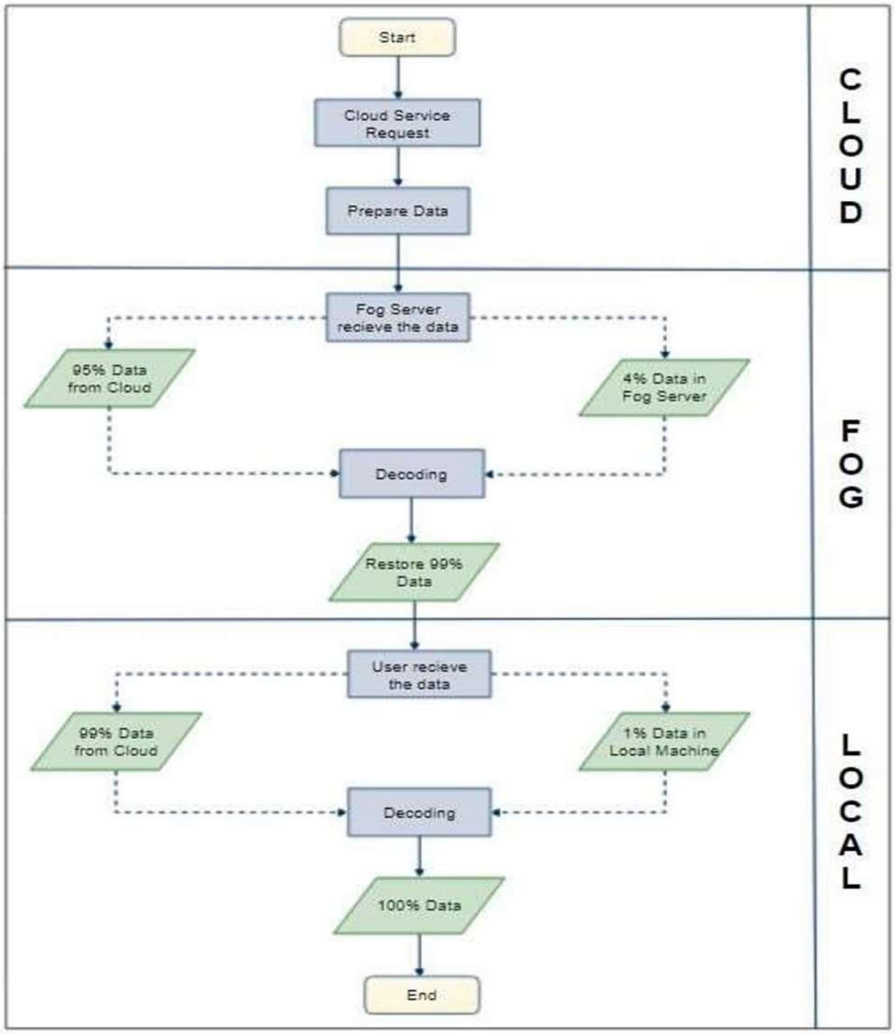

Figure 4. Download process

Hash Solomon Algorithm

In the Hash-Solomon code, we have definitions as follows:

Definition 1. Invalid Ratio: *The ratio of the number of blocks of failure data to the number of data blocks to be used in encoding. In other words, the ratio of the number of data blocks stored in the 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. Likewise, the ratio between the number of data blocks stored on the fog server and the number of data blocks stored on the cloud server.*

Definition 2. Maximal Invalid Ratio: *the maximal invalid ratio is the ratio of invalid data to the number of all data blocks when the upper server can recover the complete data by the stored data blocks. If there is one more invalid data block, the upper server can't recover the complete data anymore.*

The Maximal Invalid Ratio can be expressed as t / s+t *in the Hash-Solomon code. For convenience, we consider two layers of situations. We assume that there is* x *MB of data prepared to save. After encoding, there will* (s+t/ t)* x *data. We prepare to save* k% *in the lower server.*

The relationship between s, t *and* k *can be expressed through functional transformation as formula (2). If the parameter* k *is determined, the parameter* s *can be expressed by* t*. So, when we use our scheme, we can only upload the remaining data to the cloud server. All these operations are based on the Hash-Solomon code.*

Hash-Solomon code is a kind of coding method based on Reed Solomon's code. After encoding it by Hash-Solomon code, the data will be divided into s *parts and generate* t *redundant data. Hash-Solomon code has such property that in these* s+t *parts of data if someone has at least* s *parts, he can recover the complete data. In other words, nobody can recover the complete data with less than* s *parts of the data. According to this property of Hash-Solomon code, in our scheme, we let no more than* s-1 *parts of data be stored in a higher server with a larger storage capacity and the remainder in the lower server. In this way, the attacker cannot recover the complete data even if one of the three layers' data was stolen. Thus, we can ensure the privacy of user's data. Then, we consider the value of* s *and* t*. They are assuming that we want to save k% data on the fog server.*

$$s = \frac{(t - 2tk) + \sqrt{(2tk - t)^2 - 4t^2k^2}}{2k} \qquad (2)$$

The parameter s *is the number of blocks after data is divided, the parameter* t *is the number of redundant data blocks, and the parameter* k *is the storage ratio of different servers. Besides, the fog server includes Computational Intelligence, which can help the system calculate the results of the values of* s *and* t *because the fog server nodes have computing power.*

Working

Firstly, the user's data will be encoded on the user's local machine. Then, for example, let 1% of encoded data be stored in the machine. Then, upload the remaining 99% of the data to the fog server. Secondly, we do operations on the fog server similar to the data from the user's machine. About 4% of the data will be stored in the fog server, and the remaining data will be uploaded to the cloud server. All these operations are based on the Hash-Solomon code.

Hash-Solomon code is a coding method based on Reed Solomon's code. After encoding it with Hash-Solomon code, the data will be divided into s *parts and generate* t *redundant data. Hash-Solomon code has such property that in these* s+t *parts of data if someone has at least s parts, he can recover the complete data. In other words, nobody can recover the complete data with less than* s *parts of the data. According to this property of Hash-Solomon code, in our scheme, we let no more than* s-1 *parts of data be stored in a higher server with a larger storage capacity and the remainder in the lower server. In this way, the attacker cannot recover the complete data even if one of the three layers' data was stolen. Thus, we can ensure the privacy of user's data. Then, we consider the value of* s *and t. They assume we want to save* k% *data on the fog server.*

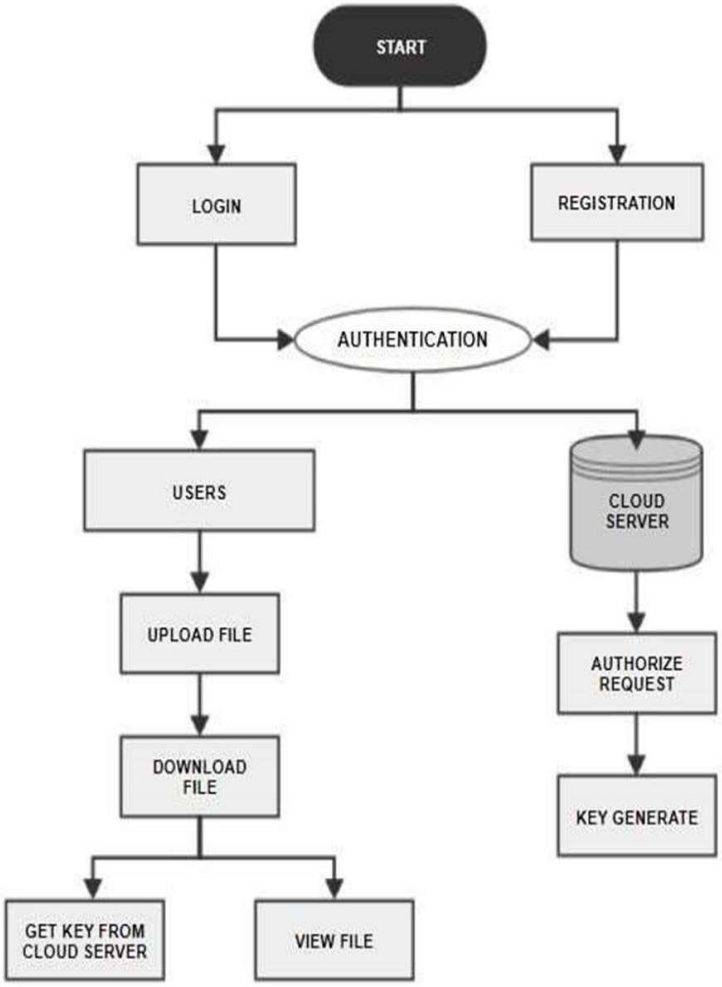

Figure 5. Workflow

Advantages

Using Reed-Solomon code, the data is encoded so that any intruder cannot retrieve the exact data stored.

By implementing Hashing, the data is stored in the split, ensuring privacy.

Fog computing serves as the middle level that includes a certain amount of data that helps encode and store the data in different layers.

5. Results

The files of different formats and sizes were uploaded and encoded by implementing the Hash-Solomon code algorithm. As shown in Fig. 5.1, results are observed to produce maximum through-put with lesser time delay, which is considered efficient.

The proposed system has proved more efficient than the existing one since we have implemented Hashing and the Reed-Solomon code algorithm for encoding and isolating the data. In addition, computer intelligence is also incorporated to partition the encoded data into different proportions and store them appropriately in local systems, fog servers, and cloud storage.

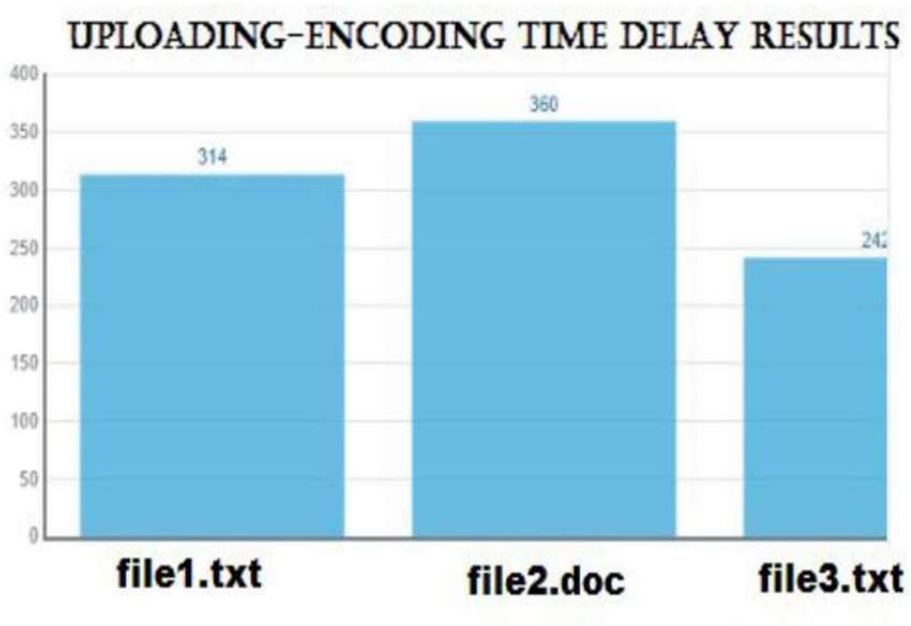

Figure 6

6. Conclusion and Future Enhancements

Cloud computing technology brings a lot of benefits to us. Cloud storage is an advantageous technology that helps users increase their storage space. Cloud storage, however, also triggers several security problems. Users do not directly monitor the physical storage of their data by using cloud storage, resulting in ownership isolation and data protection. To address the privacy issue in cloud storage, we propose a TLS architecture focused on fog computing and implementing a Hash algorithm. The system is proven viable by the theoretical safety analysis. By reasonably allocating the data block ratio stored on different servers, we can ensure data protection on each server. On the other side, the encoding matrix is technically challenging to break. In addition, the use of hash transformation can protect the fragmentary details. This scheme will efficiently fully encode and decode without cloud interference via the experiment test. In addition, we design a reasonably comprehensive efficiency index to maximise efficiency, and the Cauchy matrix is more effective throughout the programming process.

Future Enhancements

The proposed scheme is proved to be feasible through the results from experimental analysis. Additionally, some future work can be done. Along with the Hash-Solomon code algorithm, it can be combined with some other algorithms to obtain more speed and efficiency.

References

[1] Wang, Tian, Zhou Jiyuan, Chen Xinlei, Wang Guojun, Liu Anfeng, and Liu Yang. (2018). A Three-Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing. *IEEE Transactions on Emerging Topics in Computational Intelligence, 2*(1), 1-16.

[2] An, Yongkai, Lin Xiaodong, Liang Wei, Tan Gang, and Nanda Priyadarsi. (2019). A Secure Privacy Preserving Deduplication Scheme for Cloud Computing. *Future Generation Computer Systems, 13*(1), 68–96.

[3] Xia, X., Wang, L., Zhang, Z., Qin, X., Sun, X., and Ren, K. (2016). A Privacy-Preserving and

Copy-Deterrence Content-Based Image Retrieval Scheme in Cloud Computing. *IEEE Transactions on Information Forensics and Security, 11*(11), 2594–2608.

[4] Kulkarni, R., Forster, A., and Vinayagamoorthy, G. (2010). Computational Intelligence in Wireless Sensor Networks: A Survey. *IEEE Communication Surveys & Tutorials.*

[5] Li, Y., Wang, T., Wang, G., Liang, J., and Chen, H. (2016). Efficient Data Collection in Sensor-Cloud System with Multiple Mobile Sinks. In *Proceedings of the 10th Asia-Pacific Services Computing Conference.*

[6] McEliece, R. J., and Sarwate, D. V. (1981). On Sharing Secrets and Reed-Solomon Codes. *Communications of the ACM, 24*(9), 583–584.

[7] Li Hui, Sun Wenhai, Li Fenghua, and Wang Boyang. (2014). Secure and Privacy-Preserving Data Storage Service in Public Cloud. *Journal on Computer Research and Development, 51*(7), ISSN: 1397–1409.

[8] Xiao, L., Li, Q., and Liu, J. (2016). Survey on Secure Cloud Storage. *Journal of Data Acquisition and Processing, 31*(3), ISSN: 464–472.

[9] Rezapour, R., Asghari, P., Seyyed Javadi, H. H., and Ghanbari, S. (2021). Security in Fog Computing: A Systematic Review on Issues, Challenges and Solutions. *Computer Science Review, 41*, 100421.

[10] Kaur, Jasleen, Agrawal, Alka, and Khan, Raees Ahmad. (2020). Security Issues in Fog Environment: A Systematic Literature Review. *International Journal of Wireless Information Networks, 27*(3), 467-483.

[11] Ometov, A., Molua, O. L., Komarov, M., and Nurmi, J. (2022). A Survey of Security in Cloud, Edge, and Fog Computing. *Sensors, 22*, 927.

[12] Saad Khan., Parkinson, Simon., Qin, Yongrui. (2017). Fog Computing Security: A Review of Current Applications and Security Solutions. *Journal of Cloud Computing*, 6(19).