Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7759065

A THREE-LAYER PRIVACY PRESERVATION CLOUD COMPUTING USING FOG LAYER AND HASH SOLOMON CODE

[1]Abhishek M. Dhore , [2]Dr. Dinesh Kumar Sahu
[1]Ph.D Scholar,Computer Science Department, SRK University, Bhopal, India
[2]Ph.D Guide, Computer Science Department, SRK University, Bhopal, India
[1]abhishekdhore811@gmail.com, [2]dr.dineshksahu@gmail.com

ABSTRACT:
Computing tasks traditionally handled by central nodes are delegated to peripheral devices in fog computing. The explosion of unstructured data coupled with advances in cloud computing has led to the rapid improvement of cloud storage methods. Your cloud service provider makes no recommendations about the content or location of your data and information saved in the cloud. Encryption is commonly used as the foundation for privacy protection measures. A cloud-based, multi-tiered, and fog-based storage infrastructure. The suggested framework is capable of making use of cloud storage while still keeping user information secure. In this case, we employ a data-splitting technique based on the hash-Solomon code. To protect the data in this architecture, we employ methods based on the bucket concept and the BCH coding algorithm to solve the error-correcting cycle problem. This programme uses AI to determine what share of data is most effectively kept in the cloud, in the fog, and on a local system.
Keywords – Hash-Solomon code, Fog computing, cloud computing, bucket framework

INTRODUCTION: Cloud computing, in the field of computer science, refers to a form of outsourcing of computer services analogous to the outsourcing of the provision of electricity. Users need only put it to use. They are not concerned with the generation or transmission of the power they use. Their consumption is billed every month. Cloud computing is based on a similar premise; users can use resources such as data storage, processing power, and specialised application development environments without needing to understand the underlying infrastructure[1]. Computing in the cloud is often done online. Based on the way the Internet is depicted in computer network diagrams, "the cloud" is a metaphor for the Internet; this abstraction serves to conceal the Internet's complicated underlying infrastructure. This method of computing allows customers to access technology-enabled services via the Internet ("in the cloud") without needing to understand or manage the underlying servers or infrastructure.
Large cloud systems and big data architectures exhibit characteristics of fog computing, which describes the increasing challenge of gaining unbiased access to data. As a result, the information that is gathered is of poor quality. There is some uncertainty as to how fog computing will influence cloud and big data infrastructures. One thing that holds true across the board is the difficulty in delivering content precisely, and this problem has been addressed through the development of measurements meant to boost precision. The "control plane" and the "data plane" are the two basic pillars of fog networking. Fog computing, for example, enables the relocation of data-plane computing services closer to the network's edge rather than on servers in a centralised data centre[2]. Improvements in edge analytics and stream mining

can be made to enhance the quality of service (QoS) provided to customers, as well as reduce latency and save bandwidth on the backbone, fog computing places an emphasis on being closer to end users and client objectives, having a dense geographical distribution, and pooling resources locally and redundancy in case of failure.

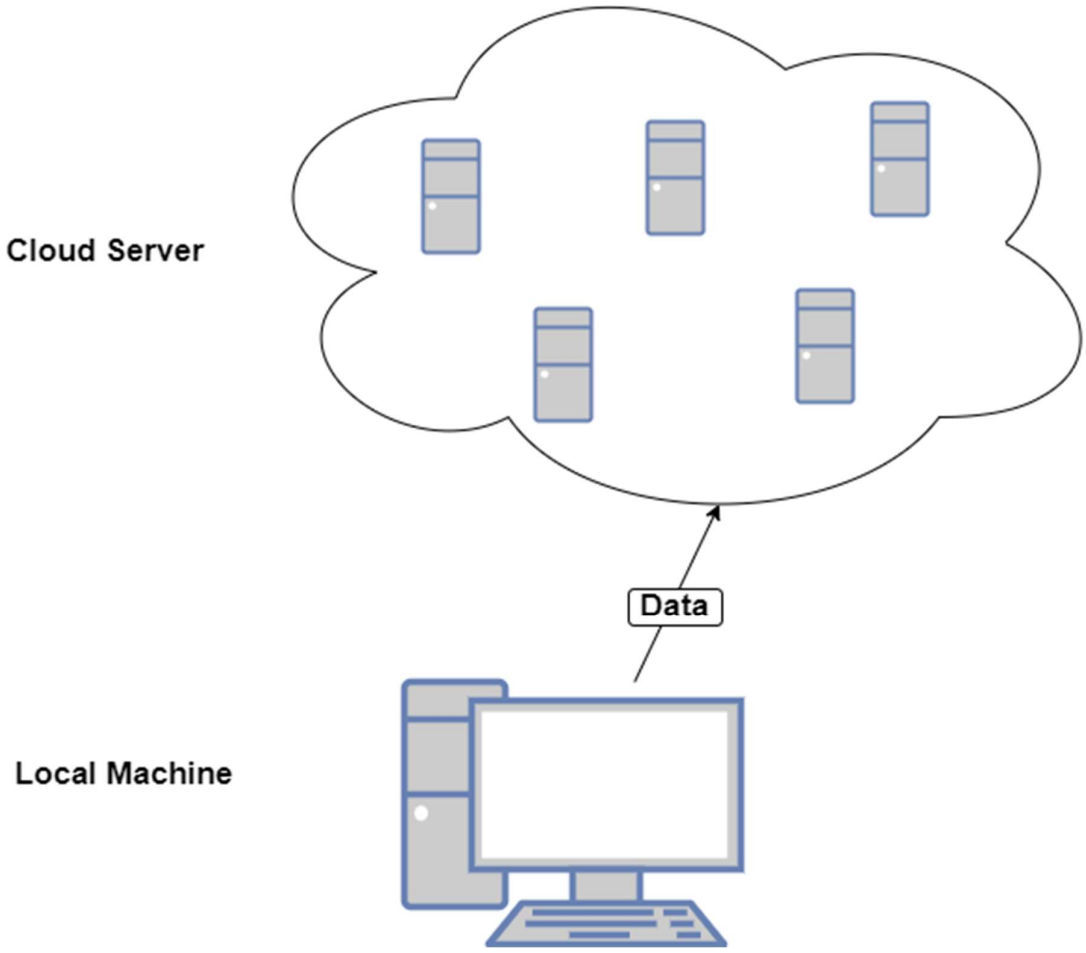

Figure 1: Traditional Cloud Storage

It can be seen in Fig. 1 that The user makes a direct connection to a cloud server and does an upload. The CSP then acts in place of the user to administer the data. Thus, users have no control over the actual location of their data storage, which might cause issues with data ownership & management continuity[3]. The CSP is not restricted in any way in their ability to view or search the cloud-based information. Meanwhile, the attackers can access the user's information by breaching the CSP server. Both of these scenarios put users at risk of having sensitive information leaked or lost. Most current approaches to securing data in the cloud centre on either restricting user access or encrypting all stored information. With these techniques, we can get rid of most of these issues. No matter how much progress is made in the algorithm, none of these approaches will be able to effectively counter an inside attack. As

a result, In this paper, we propose a fog-based TLS approach and develop a Reed-Solomon-inspired Hash-Solomon code[4][5].

Fog computing offers a broader cloud-based computing approach by utilising a huge number of fog nodes. These nodes can serve as data storage and processing hubs.

The user's information is kept secure by being split between three locations (the fog server, the cloud server, and the user's local PC) using our technology. Furthermore, the method can guarantee that the actual data cannot be rebuilt by partial data, based on the feature of the Hash-Solomon code. In contrast, Hash-Solomon encoding creates some redundancy that might be exploited during decoding. Adding more redundant blocks to a storage system can improve its reliability but will require more space to store the additional information.

Users' information is safe within our scheme due to the fair distribution of data. Computational Intelligence can be used to simplify the complex calculations required by the Hash-Solomon code (CI). Challenges in many areas, such as those found in wireless sensor networks, have been met with success through the use of CI paradigms in recent years (WSNs)[6]. Adaptive mechanisms with intelligent behaviour are made available by CI in highly dynamic settings like WSNs [9]. So, our paper uses CI to do some math out there in the fog. When it comes to interior privacy, our technique is superior to conventional ways, especially when dealing with CSPs.

Here is how the remaining portion of the paper is structured: The paper is organised as follows: Section II includes a literature review, Section III describes the TLS structure and implementations of the workflow, Section IV provides experimental assessments of the system, and Section V closes the paper.

RELATED WORK:

Both academics and businesspeople have paid close attention to the issue of cloud storage security. Many studies have been conducted over the past few years on safe cloud storage designs. Defeating the cloud's privacy problems Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, in their paper “An encrypted and watermarked CBIR strategy” was presented in A cloud-based method for retrieving images with enhanced security and copy-prevention features[7]. Using this method, photographs stored on a less-than-honest cloud server will be protected, and the image user will be dissuaded from sharing the retrieved images without proper permissions. Since Shen et al. consider the cloud to be only partially trustworthy, they advocate using attribute-based cryptography in a shared data framework for cities[3]. Their proposed system is safe, and it can withstand any attacks that might be made on it. Fu et al. boost the intelligence of semantic search by putting forth a content-aware search scheme. Their scheme has been proven effective by experimental evidence[8].

They believe that even if a CSP can be trusted, attackers can still gain access to a user's data if they gain access to the cloud storage management node. They propose using encrypted indexes and an asymmetric challenge-response authentication scheme to get around this problem. During the data request process, the user sends a password to the cloud server. To prevent password sniffing, the design uses an asymmetric response mode. Security and anonymity in a decentralised system are important to cloud storage, according to authors Fernandez et.al.[9]. So, they suggest an SSL- and Daoli-based virtual security system for online transactions. A key component of the system is the use of SSL for data transit and the installation of Daoli on the cloud server, which together encrypt information just before it is written to disc[10].

Feng argues that the research is wrong to assume that cloud servers won't experience increased server load or data leaks during transmission. Feng suggests a more streamlined approach: encrypting data on a private cloud. In addition, it supports multi-point encryption for safe data storage. This encryption, however, makes cloud searches more laborious. The ability to search encrypted data is a big topic in the cloud computing industry right now[11].
When Seny and Kristin began having doubts about the company providing their service, they decided to create a private virtual storage system based on state-of-the-art cryptographic techniques[12]. This service combines the private cloud's security and confidentiality with the public cloud's scalability and low prices to provide users the advantages of both. Since users do not have physical possession of the outsourced data, ensuring its security on the cloud can be difficult, as pointed out in the paper "Privacy-preserving public auditing for safe cloud storage" by Wang et al. Therefore, customers should be able to use a TPA to check the legitimacy of their cloud-based data if they so choose[13]. In addition to our original finding, they propose a secure public cloud storage system where the TPA can save user data so that multiple users can be audited at once. Shen et al. provide public auditing protocols that encompass global and sampling block minimum verification and batch auditing, which they argue do a better job of helping data dynamics than the existing state-of-the-art[13].

In their study "Security and privacy for storage and computation in cloud computing," Wei et al. point out that most prior attempts at cloud security focus on storage security rather than having to incorporate computation security[14]. As a result, they propose a protocol called SecCloud for preventing privacy leaks and auditing secure computations in the cloud. This ground breaking technique is the first of its type to employ designated verifier signatures, probabilistic sampling, and batch verification to safeguard confidential information from disclosure.
In their work titled "Security and privacy for storage and computation in cloud computing," Wei et al. point out that most previous studies on cloud security have focused on storage security rather than considering computation security combined. Because of this, they propose a protocol for discouraging privacy leaks and auditing secure computations in the cloud; they call it SecCloud. It's the first protocol of its kind, and it uses designated verifier signatures, batch verification, and probabilistic sampling to prevent leaks of sensitive data[14].

All the aforementioned studies aim to better secure user privacy when using cloud storage. In various roles, several of them employ various encryption strategies. Some people get around the privacy issue by auditing or constructing a security system. However, these studies all share a similar flaw. When the CSP is no longer reliable, none of these strategies will work. They are defenceless against both external threats and the CSP's unlawful data-selling practises. Since the user's data is stored in its entirety in the cloud, even the most sophisticated encryption technologies will be rendered useless once the data falls into the hands of malicious attackers.

PROPOSED METHODOLOGIES.

We propose a fog-based computing paradigm with a Three Layer Storage (TLS) framework for user privacy protection[15]. The TLS framework has the potential to grant the user some degree of control and to successfully safeguard their privacy. As was previously indicated,

fending off an assault from within is no easy task. While conventional methods are effective against external threats, they are useless when CSP itself is at fault. In contrast to standard methods, our solution uses encoding to split the user's data into three subsets of varying sizes. For reasons of secrecy, each of them will be missing some vital information. The three levels of data storage (the fog server, the cloud server, and the user's local workstation) are arranged in descending order of data size when cloud computing is paired with the fog computing idea. With this strategy, even if an attacker gains access to all data stored on a given server, he will be unable to reconstruct the user's original data. The CSP cannot get useful information without accessibility to both the user's local workstation and the fog server.

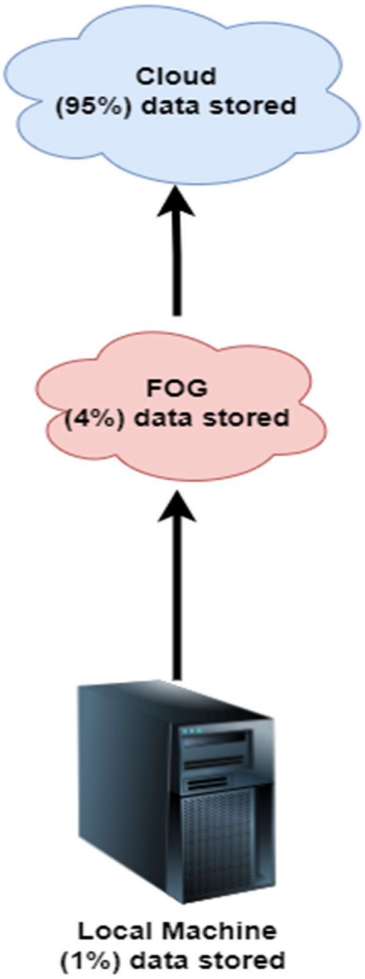

Figure 2: Three Layer System Architecture based on Fog Layer

It can be seen in Fig.2 Second, that the storage and processing power of the fog server is fully utilised by the Three Layer Security (TLS) framework. The three levels of the system are the fog server, the cloud server, and the local machine. Users' allocation strategies influence how much information is saved on each server. The first step is for the user's data to be encrypted on their own device. Then, for illustration, the computer may keep data encoded with a 1% error rate. The remaining 99 percent of the data must be sent to the fog server. Second, the user's machine data undergoes the same processing on the fog server. Around 4% of data will be kept on the fog server before the rest is transferred to the cloud. All of those processes

depend on something called the Hash-Solomon code. Hash-Solomon code, in other words, is a special case of Reed-Solomon code[16][17]. Using Hash-Solomon encoding, the data will be divided into k pieces and m redundant pieces of data will be generated. Due to the feature of the Hash-Solomon code, if a person has access to k of the m pieces of information, they may reconstruct the entire set. Alternatively stated: no one can reconstruct the full data from less than k data points. Based on this characteristic of the Hash-Solomon algorithm, our system only permits the top server (with more storage space) to hold no more than thank-1 portions of data, while the bottom server stores the remaining data. This means that even if the thief manages to take data from one of the three levels, they will not be able to reconstruct the full picture. Users' information is protected in this way.

Hash- Solomon Code Algorithm

The Hash-Solomon coding algorithm's primary function is to partition data into smaller pieces. Here, the information is stored in three distinct locations. In order to better understand the data, it is first organised by the significance of the information it contains.

BCH Algorithm

Bose-Chaudhuri-Hocquenghem codes, or BCH codes for short, are a type of cyclic error-correcting codes in coding theory. Specifically, they are constructed using polynomials over a field that is finite in size (also called the Galois field)[18]. In 1959, French mathematician Alexis Hocquenghem came up with BCH codes[19]; around 1960, Raj Bose and D. The name Bose-Chaudhuri-Hocquenghem (or the acronym BCH)[20] is derived from the initial letters of the last names of the 3 inventors (Ray-Chaudhuri, K. (mistakenly, in the case of Ray-Chaudhuri).

When designing a BCH code, the number of symbol errors it may fix is precisely under the designer's command. In particular, binary BCH codes that are capable of correcting multiple-bit errors can be designed. One more perk of BCH codes is that they are straightforward to decode using an algebraic technique called syndrome decoding. This makes it easier to create a decoder for these codes with modest, power-efficient electronics. Uses for BCH codes include two-dimensional bar codes, satellite communications, CD players, DVD players, disc drives, and solid-state drives.

In the proposed system, we had applied the BCH code for secure data transmission through the communication channel from local machine to fog layer and cloud layer. The BCH code typically was used for image and video transmission using encryption mechanism under our three layer secure structure.

MODULE DESCRIPTION

Registration Module

The user can register their login id using this module by supplying only the bare minimum of personal data. Users only want to be asked for the bare minimum of information, including an email address, name, cell phone number, and password. In order for them to be able to access the site.

Login Module

Within this section, visitors can enter their username and password to access the website. It won't let anyone in who isn't a legitimate user in there. The website is password-protected and accessible only to verified users.

Storage Module

With this component, users have the option of utilising any one of three available cloud storage services to back up their data. Information can be stored in a variety of places, including the cloud, the fog, or even on a local server. Eighty percent of our data is kept in a cloud server. Only 15% of the data in the fog server is considered very sensitive. Only 5% of the information we have is kept on the local server.

Recovery Module

Users have the option of retrieving lost data from one of three available servers. If the data conforms to these three criteria, the BCH algorithm will add it to the bucket. The user can simply retrieve their data from the bucket framework even if it has been compromised at any of the layers.

Download Procedure

Fig.5. depicts the steps an end user must take in order to retrieve his file from the cloud server. The process begins with the user making a request, which is then processed by the cloud server and integrated across several servers. Information can be stored in a variety of places, including the cloud, the fog, or even on a local server. Second, the information is transferred from the cloud server to the fog server. With the encoding key and the fog server's 5% data chunks, we can get back 100% of the original data. When this happens, the user receives 100 percent of the data from the fog server. The user finally receives the data first from fog server in step three. In order to obtain all of the information, users must
repeat the procedures outlined above.

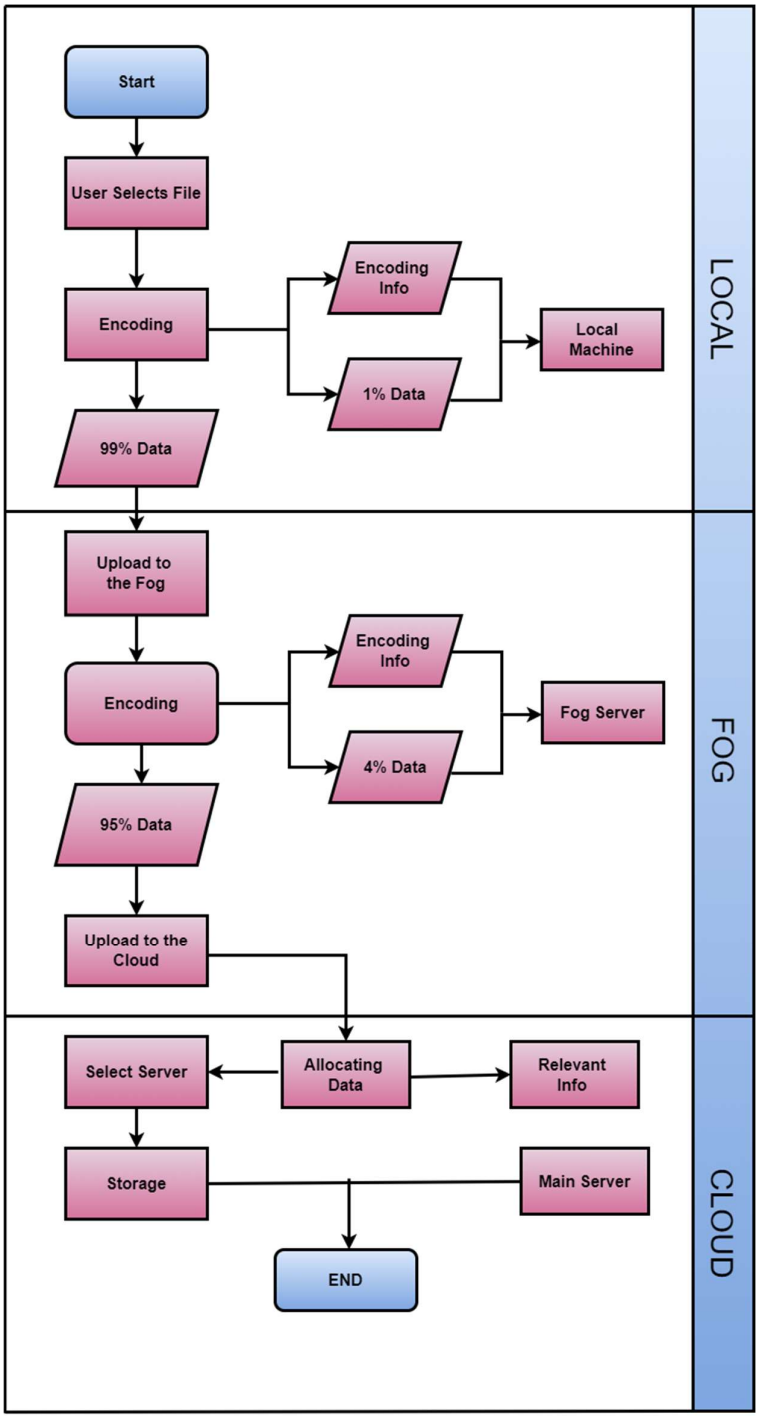

Figure 3: Stored Procedure

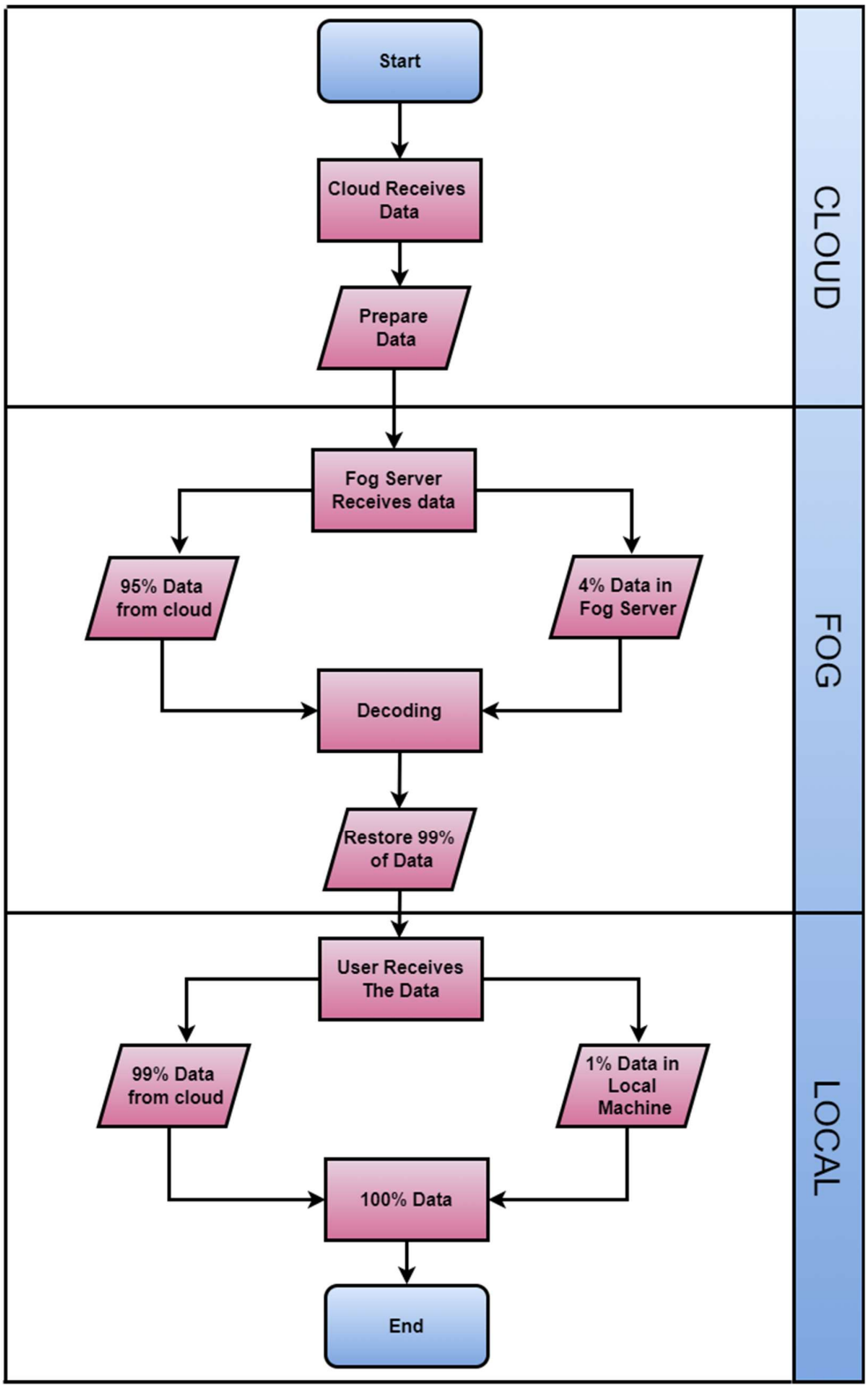

Figure 4: Download Procedure

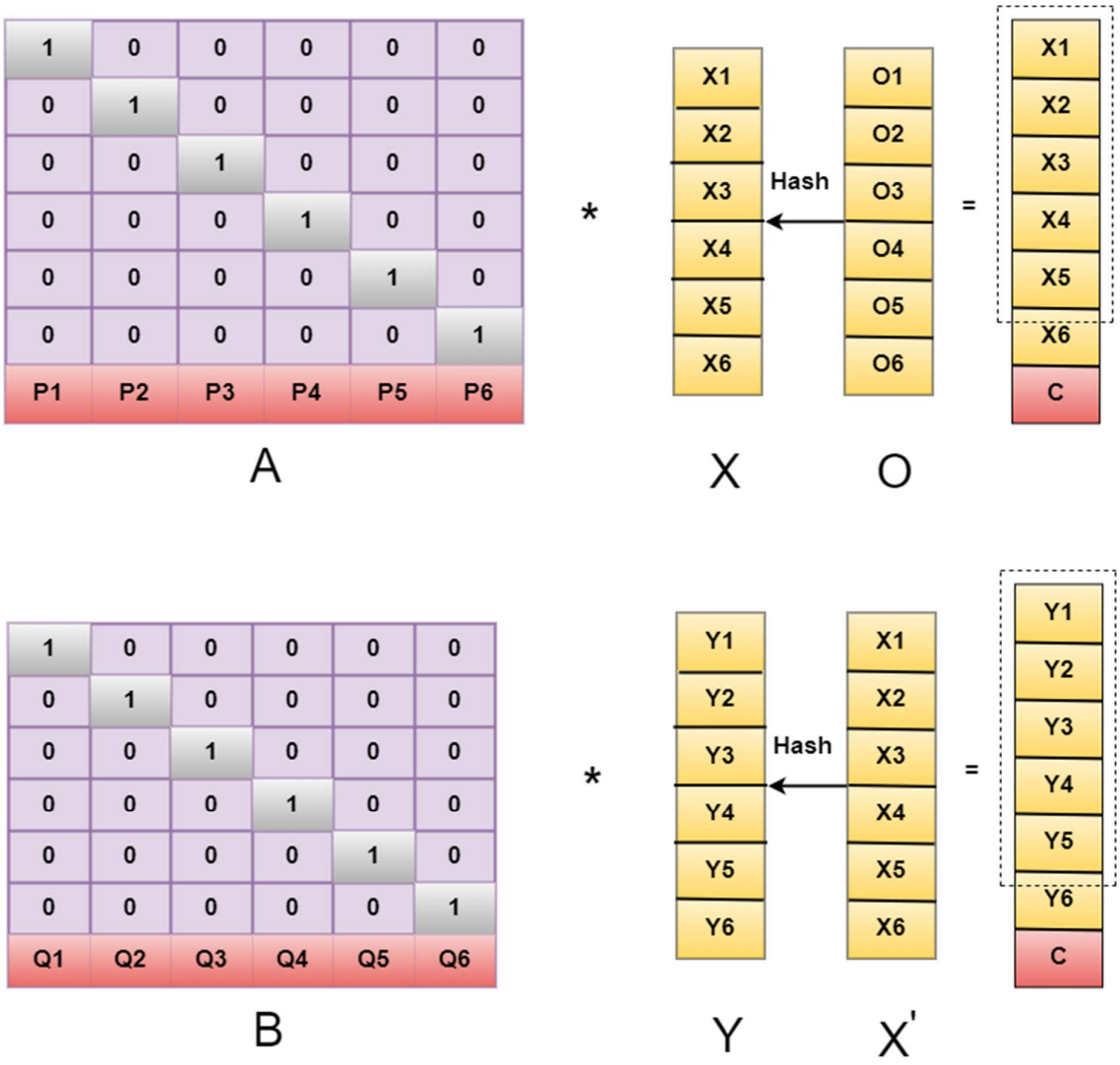

Figure 5: Download Procedure

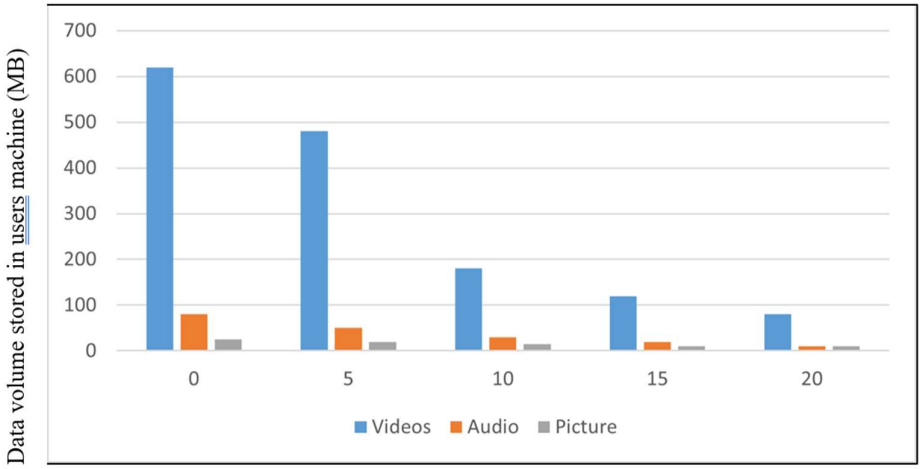

Figure 6: Results of local storage volume of different types of file

EXPERIMENT AND RESULT

This research examines the efficacy of a three-layer server framework based on the fog computing concept by taking a novel way to evaluate its performance and feasibility. This involves encoding, decoding, and testing with varied data volumes.

We show the connection between both the number of blocks and the amount of data that may be stored locally on a user's computer using a wide range of examples. Thus, M represents the number of unnecessary data blocks, while K represents the bare minimum needed by the raw data. The amount of information kept on the user's local machine is reduced as K grows. As a result, the more data there are, the less space there is on the device itself.
The efficiency of the experiment rises as the amount of data used grows. For this reason, in the real system, it is essential to increase the K value of the user's storage. The files must always be merged as with the smaller files before they can be uploaded.

CONCLUSION:

There are other upsides to using cloud services. Users are able to save larger data sets because to cloud storage. Users of cloud storage have no say over where their files are physically stored, which can lead to data loss and privacy concerns. To address the issue of data protection in the cloud, we have designed a three-tier server architecture using the fog computing concept and the hash-Solomon algorithm. The encoding matrix will be used to translate between the various storage locations (fog server, cloud server, or local machine) and the corresponding code or data. In theory, there is no way to break the encoding matrix. So, the attacker cannot get the user's information easily.

REFERENCES:

[1] N. Santos, K. P. Gummadi, and R. Rodrigues, “Towards Trusted Cloud Computing.”

[2] I. Stojmenovic and S. Wen, “The Fog computing paradigm: Scenarios and security issues,” *2014 Fed. Conf. Comput. Sci. Inf. Syst. FedCSIS 2014*, pp. 1–8, Oct. 2014, doi: 10.15439/2014F503.

[3] J. Shen, D. Liu, J. Shen, Q. Liu, and X. Sun, “A secure cloud-assisted urban data sharing framework for ubiquitous-cities,” *Pervasive Mob. Comput.*, vol. 41, pp. 219–230, Oct. 2017, doi: 10.1016/J.PMCJ.2017.03.013.

[4] C. Wang, Q. Wang, K. Ren, and W. Lou, “Ensuring Data Storage Security in Cloud Computing.”

[5] S. Asif, U. Veerash, and G. K. L. Kumar, “A system of privacy preserving public auditing for secure cloud storage system,” *AIP Conf. Proc.*, vol. 2358, no. 1, p. 100020, Jul. 2021, doi: 10.1063/5.0058354.

[6] S. B. Kulkarni, A. Förster, and G. Venayagamoorthy, “A Survey on Applications of Computational Intelligence for Wireless Sensor Networks,” *IEEE Commun. Surv. &amp Tutorials*, vol. 13, p. 25, Jan. 2011.

[7] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, “A Privacy-Preserving and Copy-Deterrence Content-Based Image Retrieval Scheme in Cloud Computing,” *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 11, pp. 2594–2608, Nov. 2016, doi: 10.1109/TIFS.2016.2590944.

[8] 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 Secur.*, vol. 12, no. 8, pp. 1874–1884, Aug. 2017, doi: 10.1109/TIFS.2017.2692728.

[9] M. Fernández, J. Jaimunk, and B. Thuraisingham, *Privacy-Preserving Architecture for Cloud-IoT Platforms*. 2019.

[10] N. Kalbo, Y. Mirsky, A. Shabtai, and Y. Elovici, "The Security of IP-Based Video Surveillance Systems," *Sensors*, vol. 20, p. 4806, Aug. 2020, doi: 10.3390/s20174806.

[11] H. Yan and W. Gui, "Efficient Identity-Based Public Integrity Auditing of Shared Data in Cloud Storage with User Privacy Preserving," *IEEE Access*, vol. 9, pp. 45822–45831, 2021, doi: 10.1109/ACCESS.2021.3066497.

[12] A. Dwivedi, "Fog Computing Application for Cloud Servers," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 7, pp. 3146–3151, May 2019, doi: 10.22214/ijraset.2019.5520.

[13] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 362–375, 2013, doi: 10.1109/TC.2011.245.

[14] L. Wei *et al.*, "Security and privacy for storage and computation in cloud computing," *Inf. Sci. (Ny).*, vol. 258, pp. 371–386, Feb. 2014, doi: 10.1016/J.INS.2013.04.028.

[15] X. Zhao, J. Zhu, X. Liang, S. Jiang, and Q. Chen, "Lightweight and integrity-protecting oriented data aggregation scheme for wireless sensor networks," *IET Inf. Secur.*, vol. 11, no. 2, pp. 82–88, 2017, doi: 10.1049/iet-ifs.2015.0387.

[16] A. Md, "A THREE LAYER PRIVACY PRESERVING CLOUD STORAGE SCHEME BASED ON COMPUTATIONAL INTELLIGENCE IN FOG COMPUTING," *Int. J. Eng. Sci.*, vol. Vol 9, Jan. 2018.

[17] H. Xu and D. Bhalerao, "Reliable and secure distributed cloud data storage using reed-solomon codes," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 25, no. 9–10, pp. 1611–1632, 2015, doi: 10.1142/S0218194015400355.

[18] C. J. Benvenuto, "Galois Field in Cryptography," pp. 1–11, 2012.

[19] "Alexis Hocquenghem - Wikipedia." https://en.wikipedia.org/wiki/Alexis_Hocquenghem (accessed Sep. 28, 2022).

[20] J. L. Massey, "Step-by-step decoding of BCH codes," *IEEE Trans. Inf. Theory*, vol. 11, no. 3, pp. 3–8, 1965.