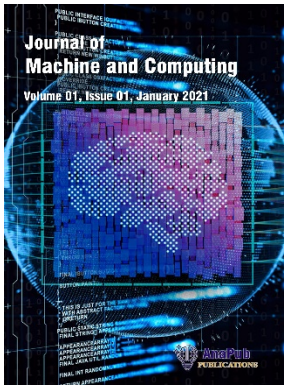
Journal Pre-proof



Enhancing Cloud Data Deduplication with Dynamic Chunking and Public Blockchain

Richa Arora, Vetrithangam D

DOI: 10.53759/7669/jmc202404050

Reference: JMC202404050

Journal: Journal of Machine and Computing.

Received 11 July 2023; Revised form 15 March 2024; Accepted 22 May 2024.

Please cite this article as:

Richa Arora, Vetrithangam D, “Enhancing Cloud Data Deduplication with Dynamic Chunking and Public Blockchain”, Journal of Machine and Computing. (2024). Doi: https:// doi.org/10.53759/7669/ jmc202404050

Enhancing Cloud Data Deduplication with Dynamic Chunking and Public Blockchain

[1]Richa Arora, [2]Dr. D. Vetrithangam
[1,2] Chandigarh University
[1]richaaeri9988@gmail.com, [2]vetrigold@gmail.com

ArticleInfo
Journal of Machine and Computing (http://anapub.co.ke/journals/jmc/jmc.html)
Doi : https://doi.org/10.53759/7669/jmcXXXXXXXX
Received xx December xxxx; Revised form xx December xxxx; Accepted xx December xxxx
Available online xx January xxxx.

Abstract

The majority of cloud service providers (CSPs) store and remove customer data according to certain principles. The majority of them have designed their cloud platform to have very high levels of consistency, speed, availability, and durability. Their systems are built with these performance characteristics in mind, and the requirement to ensure precise and rapid data deletion must be carefully balanced. In the public blockchain, this paper suggests employing the rapid content-defined Chunking algorithm for data duplication. Acute data is frequently outsourced by individuals and organizations to distant cloud servers since doing so greatly reduces the headache of maintaining infrastructure and software. However, because user data is transmitted to cloud storage providers and stored on a remote cloud, ownership and control rights are nonetheless separated. Users thus have significant challenges when attempting to confirm the integrity of private information. According to the experiment results, the suggested dynamic chunking has a fast processing time that is on par with fixed-length chunking and significantly improves deduplication processing capability.

Keywords: cloud service providers (CSPs); chunking algorithm; data duplication; blockchain;

I. INTRODUCTION

It is common knowledge that big data has arrived in modern society and that information of all kinds is rapidly expanding. The study paper states that IDC projects exponential growth in the global market's storage space from 43 ZB to 273 ZB between 2018 and 2025. The ocean is entering the big data era, making up around 80% of the planet's surface. Revolutionary advances have been made to marine observation equipment in recent years. Compared to other industries, the marine data volume is growing at a faster pace, and the volume of marine data as represented by satellite remote sensing data is significantly increasing. When considering data deletion, compression technology is typically the first thing that comes to mind.

However, string matching—more specifically, a string-matching algorithm and its variants—is used by decompression technology to reliably return the same information block [1]. More intricate but incredibly accurate and efficient for fine-grained duplication removal is the precise match of the information block's fingerprint; data deduplication technology locates the same information blocks. In this day and age, data duplication—a well-known and widely used storage technology—can efficiently maximize storage capacity. Data replication can therefore have several useful advantages, including the ability to efficiently manage data with a rapid growth rate, increase the amount of storage space that is available and improve storage efficiency, reduce overall storage and administration costs, and satisfy ROI, TCO, etc. A resulting chunk is the portion of the data block that is between its previous and current cut-points. The following steps make up the proposed structure, which is depicted in Figure 1.1.

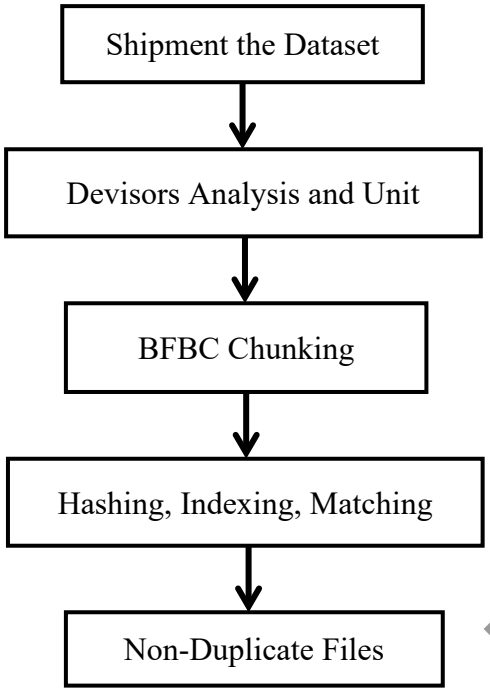


Fig 1.1 The suggested structure

The data owners no longer have control over their data once it is saved on an online platform. While there are many benefits to modern technology, it also brings with it new security risks, particularly those about data integrity. One of the most important components of any system is information integrity [2]. The information owner should activate auditing procedures to guarantee the accuracy of data that is outsourced. An external or internal auditor will conduct an audit, which is a process of study and verification, to find a system's security flaws. In this paper, we verify the data accuracy of the outsourced data through the auditing procedure.

Storage capacity is the second crucial need when it comes to user data storage. The best option for guaranteeing data storage efficiency is data deduplication. Data deduplication, also known as intelligent compression or single-instance storage, removes duplicate data and maintains a single copy of each file either before or after the data is uploaded to be kept in the cloud. By avoiding storing numerous copies of the same data, this strategy lowers data quantities and, consequently, space overhead. To get around this problem, private data such as financial and health information is kept safe on permissioned blockchains. In a permissioned network, the owner of the data maintains complete control over the network; only members who have registered can join and use the data within. In contrast to permission-less networks, permissioned networks offer users sensitive data confidentiality and integrity [3]. In cloud storage methods, authorized users' access is occasionally blocked by cloud service providers.

Decentralized storage based on blockchain technology solves this disadvantage. With little effort, authorized users can obtain data from an authorized connection. Thus, blockchain technology is used to achieve data availability. The remainder of the piece is structured as follows: In addition to discussing the current methods and plans, section 2 provides an overview of the article's purpose and value. The background ideas behind the suggested secure data deduplication method are covered in section 3. Section 4 provides protocol design models and security analysis on the suggested protocol and approach. The performance analysis of the simulation findings is covered in section 5 where we wrap up the paper.

II. RELATED WORKS

The first and most well-known decentralized cryptocurrency that enables online payments without the use of a banking system is called Bitcoin [4]. The Bitcoin system uses blockchain to address the issue of double spending. With no centralized authority, blockchain is a publicly distributed transaction ledger that is constantly growing and capable of recording an unchangeable history of events as well as offering a tamper-proof ledger. A block on the blockchain is made up of the following data components: versions, the nonce, hashing of the preceding block, the Merkle tree root encode, and date. Once the legitimacy has been confirmed, the transaction can be added to the blockchain. As a result, blockchain technology can effectively validate past data and prevent chained block modifications.

To obtain data redundancy, data deduplication technology is developed based on a comprehensive examination of privacy big data. The unique data objects that are exhibited are the result of processing duplicate data in big data, which replaces other data samples that exhibit significant levels of similarity [5]. Preprocessing technology allows for the effective removal of unnecessary data from files, hence reducing both the data dimensions and storage space. The data deduplication technique optimizes the amount of storage space for information, hence lowering the processing workload for encryption and increasing the effectiveness of large-scale data update protection.

As with most JPEG encoding methods, loss reduction eliminates information by identifying and permanently removing redundant data. Because conventional storage systems require lossless compression, this article focuses on continuous bending [6], namely data deduplication. Data compression techniques that date back to the early 1900s describe byte duplication using probabilistic designs, or stochastic encoding. By employing a frequency-sorted binary tree to produce the optimal codes for volatility coding, the Huffman entropy codification method is the most often used. To achieve higher compression and satisfy the basic information compression ratio restriction, ELIA's first proposed arithmetic code from the 1960s splits the entire data into many integers. The input is divided into different representations, one of which has a lower code, by the Huff coding method. Many smart data management methods, including virtualization, which is thin supplies, data deduplication, and hierarchy preservation, have been implemented extensively in cloud data centers to increase storage capacity utilization [7]. Deduplication is currently a hot research topic because it effectively reduces bandwidth use, saves a significant amount of storage capacity, improves read and write data performance and helps customers save energy and reduce emissions and operating costs. Finding and eliminating redundant data is therefore essential to decreasing the amount of information stored in the system. The high-performance and reliable chunking approach is the cornerstone for its detection.

The data is read in a byte stream using the data chunking technique [8]. A single byte or several bytes are chosen as the chunk boundary during the reading process based on specific criteria. A chunk is a set of data that lies between two ends of neighboring boundary ends. Chunking techniques are extensively employed in numerous domains, including recognizing texts, communication over networks, information synchronization, preservation, and caching. For instance, in the text recognition sector, a phrase must be chunked to gather the topic, prediction, thing, and other important elements before grammar analysis can determine the sentence's actual significance. A combination of deduplication techniques and convergent encryption is utilized to address the key problems associated with storing data on the cloud: privacy, safety, anonymity, and quality [9]. With classical encryption, deduplication is not possible since multiple users would encrypt the same data using different keys, producing separate cipher texts for the same information. Due to its ability to enable deduplication and guarantee data secrecy, convergent encryption is utilized. Data encryption is accomplished by computing the data copy's cryptographic hash value, which is then utilized as a converging key.

The marker could be as basic as a full stop or a series of bytes so that a particular bit pattern is produced by an equation of the data. Secondly, it periodically moves a fixed-size window, starting from the first byte of the file. Because of its great efficiency, Rabin's fingerprint method is frequently used to calculate fingerprints. This chunking border is defined here [10] each time the signature matches the marker; if not, the window shifts one byte, and a new fingerprinting is produced. Finally, use MD5 or SHA-1 to compute the encryption key between the two borders and decide if this section is redundant. Because simple sliding window chunking is used, the changes only affect the modified chunks—the remaining chunks are left unchanged. This technique yields variable-sized chunks but greatly increases the rate at which duplicates are identified.

III. METHODS AND MATERIALS

The background of chunking algorithms, their drawbacks, and the reasons behind our work are covered in this section. To get rid of duplicate data and cut down on the amount of storage needed, data deduplication is necessary [11]. It has been getting more and more attention as it has changed to satisfy the need for lower storage costs, faster backups, and less data sent across a network. In this study, we proposed a novel hash method based on a theoretical triple hashing function, as well as a CDC algorithm depending on the rate of bytes appearance.

III.i) Definition-Based Chunking

When a predetermined breaking condition is met, content-defined chunking, the chunking breakpoint is determined by a data deduplication chunking technology based on the data's contents. It is employed to address the boundary shifting problem caused by corrected size chunking, as any modification to the data flow, even a byte increases or elimination, results in the creation of a new set of chunks with distinct hash values, which means the information is deemed new and reduces the effectiveness of deduplication. The most popular content-defined chunking by deduplication technologies is listed below.

a) Sliding Window Basic

One of the older CDC chunking methods uses the Rabin fingerprint as a fingerprint hashing function as a breaking condition after defining three chunking parameters (the window's slide size).

b) Chunking with Two Threshold Divisors

The major divisor value of this technique is estimated and unrelated to the data's substance, which is a disadvantage. Furthermore, the additional divisor is typically identified very near the Tmax threshold, and the procedure will not use it until it reaches the Tmax threshold. These pointless comparisons and calculations cost a lot of computational power and degrade algorithmic speed.

c) The Suggested Frequency-Based Chunking of Bytes

A new CDC algorithm, as shown in Algorithm 1 [12], which describes the chunking algorithm, uses content analysis with mathematical models to generate a histogram of the sounds of the appearance of pair-bytes (pair-bytes shipping evaluation) in the dataset. This allows for the construction of a set of divisions to be used as chunking breaking scores.

Algorithm 1: Chunking Algorithm

Impartial: Based on Tmin, List of Divisions, and Tmax, divides the file into segments.

Effort: Any kind or size of file

List of Divisors, Tmin, Tmax

Productivity: Number of bits with varying sizes

Establish Pbreakpoint ↞ 0 and Breakpoint ↞ 0.

Read the file as a byte array
Set File Size ← Length, If the length is zero

Proceed to Step 2. Set File Size ↞ Full Length

If Length is less than or equal to Tmin

Send it to the Double hash operation, Length = Full Length—chunk length, treating it as a chunk.

Find the pair divisors pixels from breakpoint + Tmin till Full Length if Length is between Tmin and Tmax.If discovered. Think about the portion from Pbreakpoint up to the place of the divisor byte.

Send the segment to the double hash function, with the formula: length = full length − breakpoint, breakpoint = chunk length +1.

Should Length match Tmax
Think about the section from Pbreakpoint to Full Length?
Forward the chunk to Triple Hashing functions.
Chunk Length + 1 = Breakpoint – Full Length – Breakpoint

If Length Exceeds Tmax

From breakpoint + Tmin to Pbreakpoint + Tmax, look for the pair divisors bytes. If discovered, Think about the portion from Pbreakpoint up to the breakpoint.

Pbreakpoint = breakpoint is the quadruple hash function to which the chunk should be sent.

Length = Whole Length—point of break

Proceed to step 3.

The chunking throughput decreases those results from Rabin's rolling hashing function performance bottleneck affects both TTTD and BSW. Instead of calculating the Rabin fingerprints for each cut-point judgement, BFBC uses a list of divisors produced from the statistical characteristics of the information set it, which considerably improves chunking performance and improves compression effectiveness.

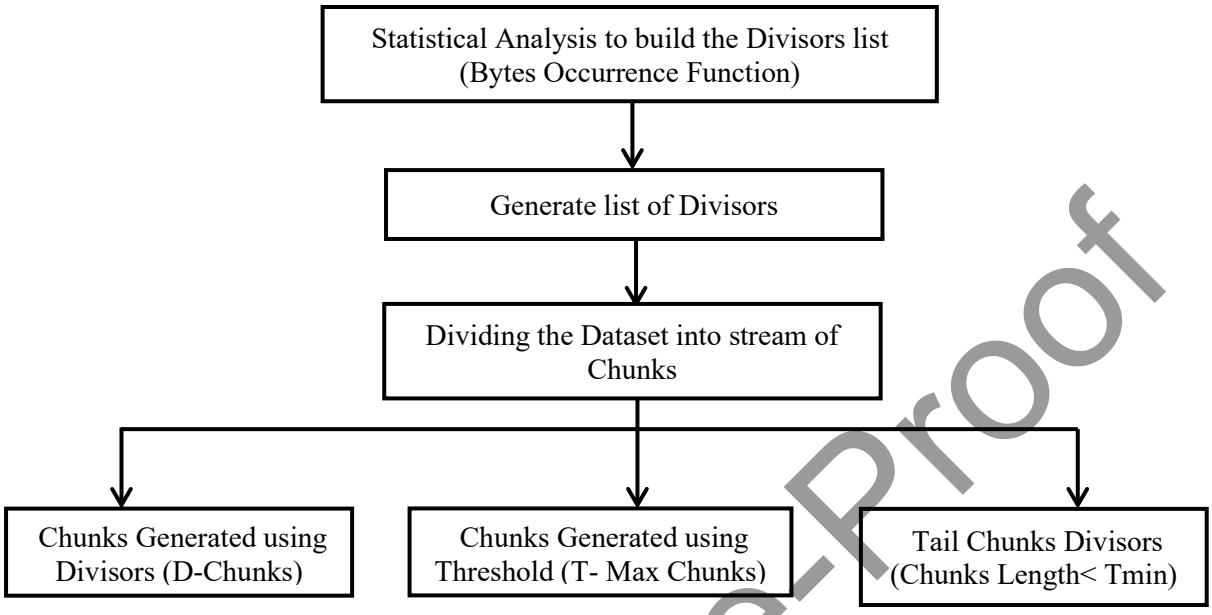


Fig 3.1 chunk types produced by the chunking phase

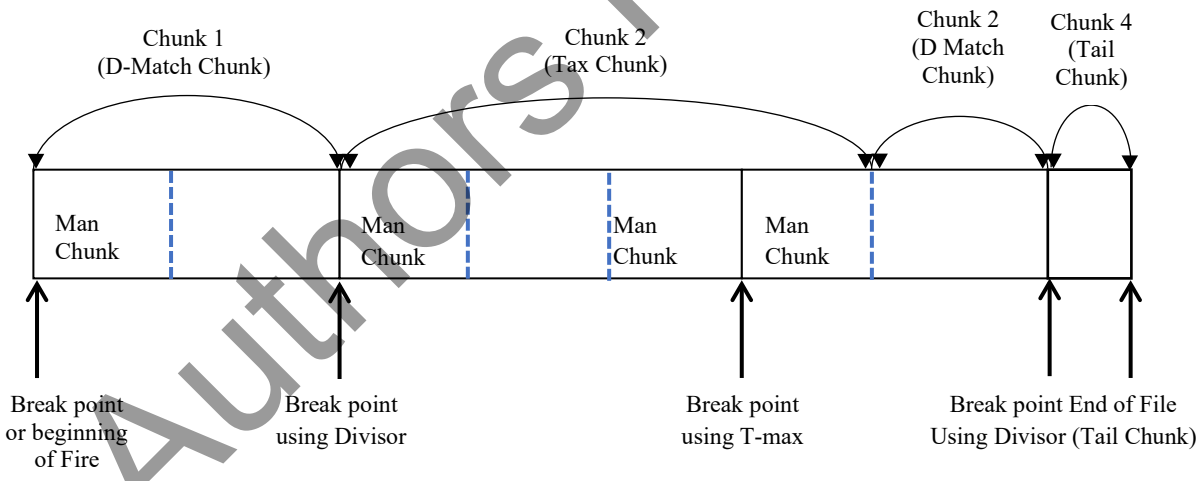


Fig 3.2 Chunking method of the suggested architecture

Figure 3.1 shows the different chunk types that are generated during the chunking stage. Figure 3.2 illustrates the chunking technique used in the proposed system [13]. Applying the Dynamic Chunking Algorithm to blockchain-based data deduplication has the following major benefits: Efficiency: By decreasing redundancy and enhancing overall performance, the algorithm's capacity to dynamically modify chunk sizes based on data attributes might result in more effective storage and retrieval procedures. Security: Data deduplication can take use of the immutability and decentralisation inherent in blockchain technology, which helps to ensure the integrity of the deduplicated data. Cost-effectiveness: When data deduplication reduces the amount of storage needed, it can save money, especially in situations where storage expenses are a major consideration. Scalability: Because the blockchain network can process a lot of transactions quickly, blockchain-based data deduplication can grow to handle increasing data volumes with ease.

3.ii) De- duplication's Security Risks

Despite significant advancements in de-duplication, several challenges remain related to strategies for removing redundant data, including attacks that jeopardy the privacy of content hosted on cloud services. Several well-known attack models that take advantage of storage service providers' data de-duplication are examined as a result of removing redundant data.

a) Make file content visible

By uploading the files in plain text to the cloud and subsequently moving them to an insecure channel, this kind of attack assists the assailant in learning the contents of the files. The hacker will create many copies of a file prototype that has every potential value that could be exclusive to a particular user, revealing the contents of the file. To presume the existence of a file, the assailant can brute force fill out the file contents and submit each version to the cloud storage service. After a value is identified as having been previously submitted, the attacker gains access to the version of the value that was entered.

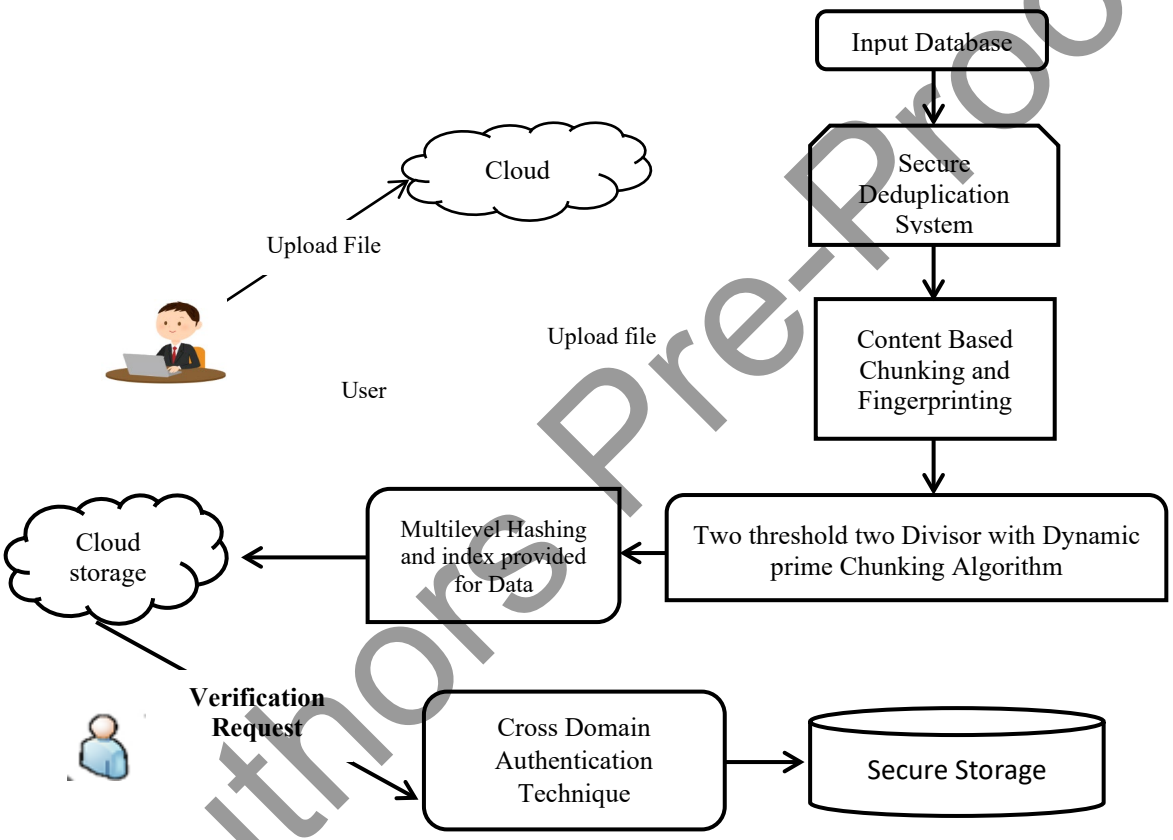


Fig 3.3 Data deduplication with Chunking-Based cloud storage

These data deduplication techniques are typically applied to identify redundant data from huge datasets, such as backup, primary, and archive storage. The main achievement of the data deduplication approach is the backup file structure, it cuts down on the logical backup data's size from a massive amount of disc space.

3.iii) Data deduplication with Chunking-Based Technique

In the chunking-based deduplication technique, each file that needs to be stored is first divided into a number of separate pieces, or "chunks." Every chunk is a single, autonomously processed string of consecutive bytes from the file. The literature contains a variety of methods for creating pieces. Among these, "static chunking" and "content-defined chunking" are the two important chunking techniques.

a) Chunking Static

Rather than using the full file as the smallest unit, the static chunking technique breaks a file into evenly sized sections, with each chunk's size set by the numerous of a disc sector or the block size of the file structure. For these reasons, fixed-sized splitting, fixed splitting, fixed chunking, and fixed chunking are other names for static chunking. The operation of the static chunking technique is depicted in Figure 3.3 [14]. Although it is the quickest chunking technique for identifying duplicate blocks, static chunking's efficiency is unsatisfactory. However, the static chunking approach causes "Boundary Shifting," a severe issue.

b) Problem of Boundary Shifting

Because of the changes in the data, boundary shifting is a major effect of fixed-size chunking. Even though the majority of the information in the file remains unaltered, fixed-size chunking would produce different outcomes for the following chunks when the user added or removed a byte from the information. We refer to this issue as the border-shifting problem. If a new byte is added to the start of the document in the following backup, the hash value of each of the subsequent chunks may change. Anything will therefore be regarded as new data and is stored on the disc as such.

c) Chunking Based on Content

Content-defined chunking is commonly used in backup systems due to its resistance to the borders shifting effect, which reduces the amount of duplicate data found by data deduplication structures. When the same data is saved more than once with a slight change, like when new data is introduced to the data stream, this effect occurs. Because the chunks in these cases are not similar, static chunking was unable to detect the redundancies; on the other hand, content-defined chunking adjusts the chunking with the material and generates the same chunks as before. The CDC technique establishes breakpoints at which a particular condition becomes true rather than defining limits at numerous of the specified chunk size. A fixed-size overlapped sliding window is typically used for this. Content-based chunking is another name for variable-size chunking. Stated differently, chunking algorithms identify file content-based chunk boundaries. The majority of deduplication systems attain high data deduplication throughput by employing the content-defined chunking algorithm.

d) Simple Sliding Window

The first chunking method to apply the working version of the hash-breaking (nonoverlap) technique is the basic sliding window (BSW) technique, which has previously been shown to provide the best results in balancing capacity expansion and deduplication ratios. Even though the majority of the breaking points in the current experiment are found at the highest threshold, this method requires additional computations to find the breakdown points from the beginning to the specified threshold.

e) Chunking with Two Threshold Two Divisor

This TTTD algorithm employs the same concept as the BSW algorithm, but to avoid the additional processing time required by the BSW to calculate the breakpoints up to the maximum threshold, it uses four parameters: the lowest threshold, the maximum threshold, the primary divide, and the additional divisor. With this method, extremely large chunks are eliminated using the lowest threshold, and very small pieces are eliminated using the maximum threshold. The second divisor result is half of the main divisor, and the main divisor functions in the same way as the BSW chunking algorithm's divisor to identify the breakpoint. If the technique is unable to locate the breakpoint using the primary divisor, the second divisor in the two criterion two divisor procedure is employed to determine the breakup point.

Until it hits the maximum threshold, the TTTD algorithm determines whether to employ the backup breakpoint discovered by the second divisor. Additionally, the breakpoints identified by the subsequent divisor are closer to the maximal threshold and greater in size.

f) Chunking-S with Two Threshold Two Divisor

To avoid the higher chunk size found by the second divisor of two thresholds two divisor techniques, a new parameter, the mean of the maximum and lowest thresholds is incorporated in the TTTD-S technique. Upon exceeding this average threshold, this approach moves half of the secondary divide value to the third divisor and the extra divisor value to the main division after the chunk size from the last delay to the current pointer is reached. The initial values of the main and second divisors are reverted to after the breakpoint has been identified. This will prevent needless computations and compares while also reducing large-size chunks.

IV. IMPLEMENTATION AND EXPERIMENTAL RESULTS

We assess the suggested approach's encoding and decoding efficiency in terms of creating and retrieving key shares, respectively. We conducted all of our trials on a Windows OS-powered Intel Xeon E5530 (3.40 GHz) [15] server. Using the traditional approach, we first assess a few fundamental modules that are included in the suggested approach:

35.196 microseconds on average are needed to generate a 32-byte hash from a 5 KB data block.

A 5 KB data block may be encrypted with a 32-byte hash on average in 33.518 seconds.

Decrypting a 5 KB data block using its 32-byte hash takes an average of 32.683 seconds.

Table 1. Distinct time values are associated with encryption

Computational Encryption Time				
Input Files	**Novel Block chain based Secure De-duplication Authentication scheme (NBSDAS)**	**Convergent Encryption**	**Efficient And Reliable Convergent Key Management**	**Secure Distributed De-Duplication System**
20	2.57	3.343	4.425	4.353
30	4.43	5.302	6.416	7.402
40	5.473	6.401	7.29	9.339
50	7.446	8.445	9.393	11.346
60	sss.473	9.334	11.336	17.346

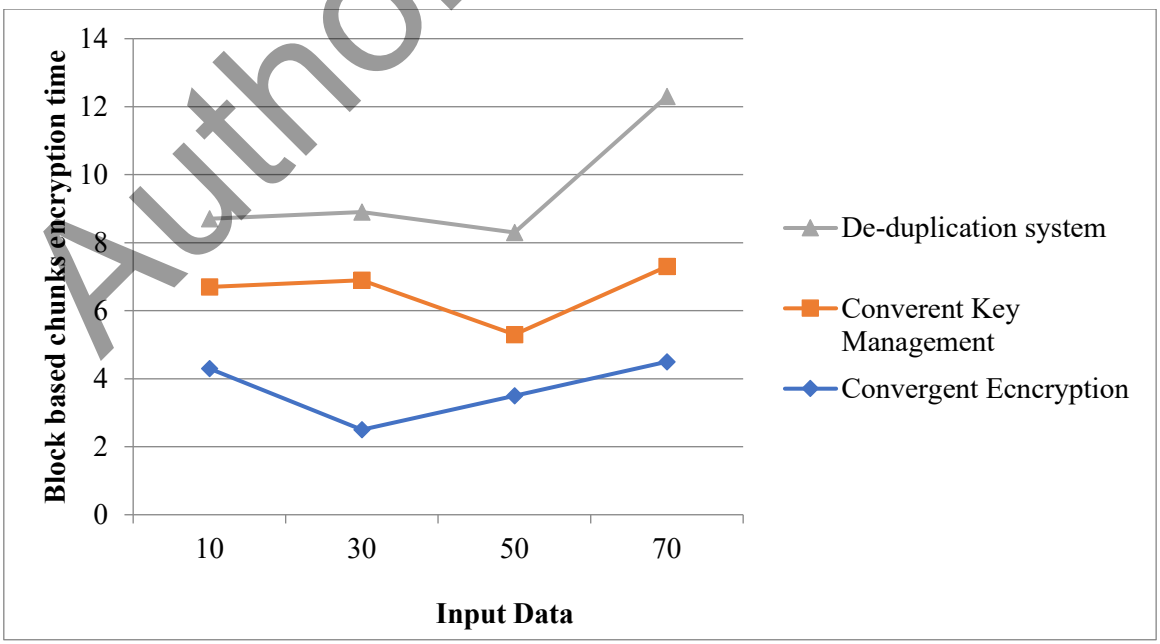


Fig 4.1 An assessment of decryption time performance for dynamic file de-duplication

Table 2. Total Execution Values for Processing Various Documents

Computational Execution Time				
Input Files	**Novel Block chain based Secure De-duplication Authentication scheme (NBSDAS)**	**Convergent Encryption**	**Efficient And Reliable Convergent Key Management**	**Secure Distributed De-Duplication System**
20	8.57	9.343	11.43	12.36
30	9.43	12.02	14.42	16.42
40	12.73	16.01	16.29	19.82
50	14.46	19.56	23.40	22.35
60	16.73	22.45	28.34	29.35

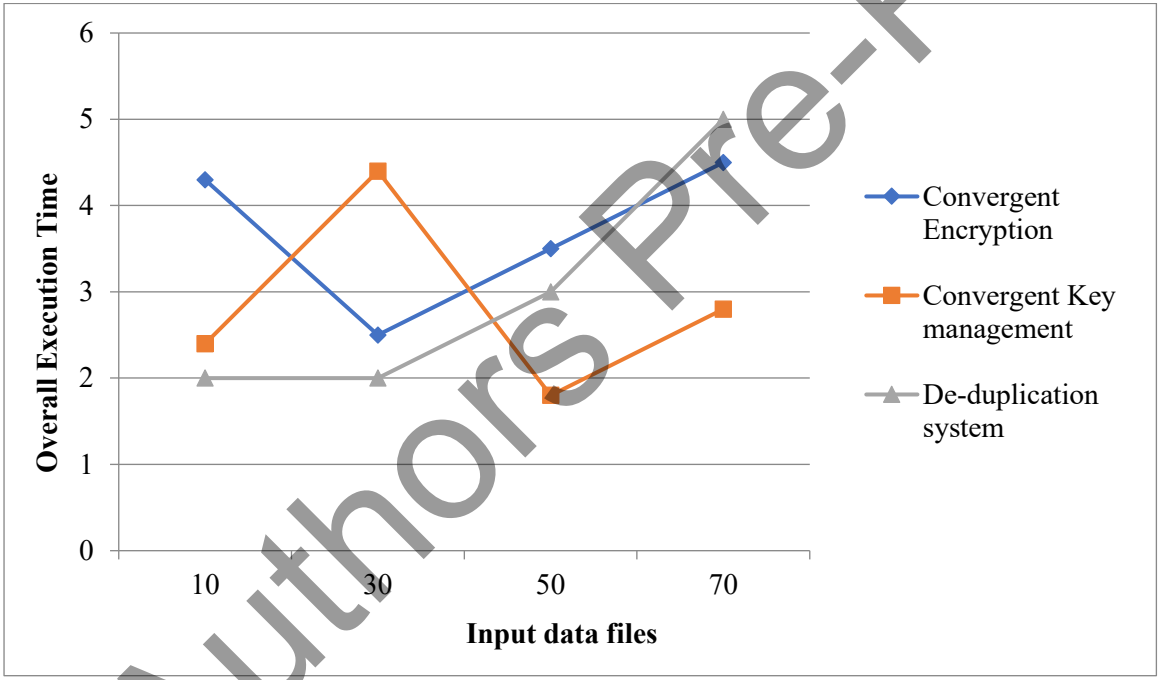


Fig 4.2 An assessment of the overall processing system's performance

Table 3. Calculating CPU processing costs for de-duplication using computation

Computation CPU Processing Cost				
Input Files	**Novel Block chain based Secure De-duplication Authentication scheme (NBSDAS)**	**Convergent Encryption**	**Efficient And Reliable Convergent Key Management**	**Secure Distributed De-Duplication System**
20	268	564	957	1054

30	457	1127	1257	1864
40	965	1569	1797	2170
50	1126	1864	2257	2957
60	1366	2676	3569	4257

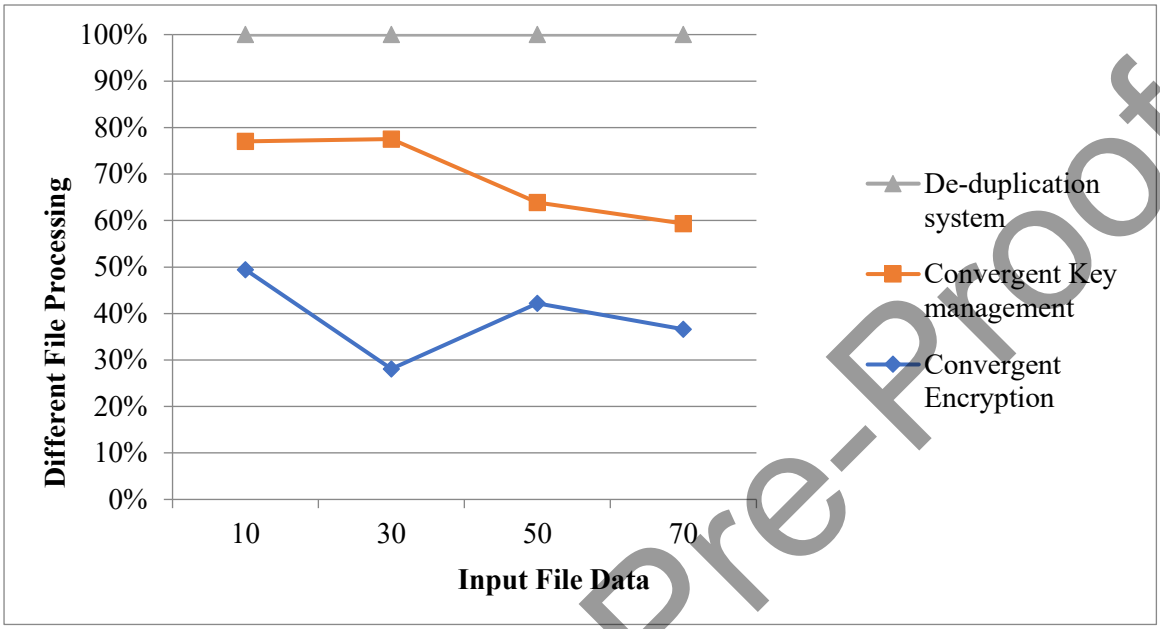


Fig 4.3 Evaluation of memory use performance during de-duplication

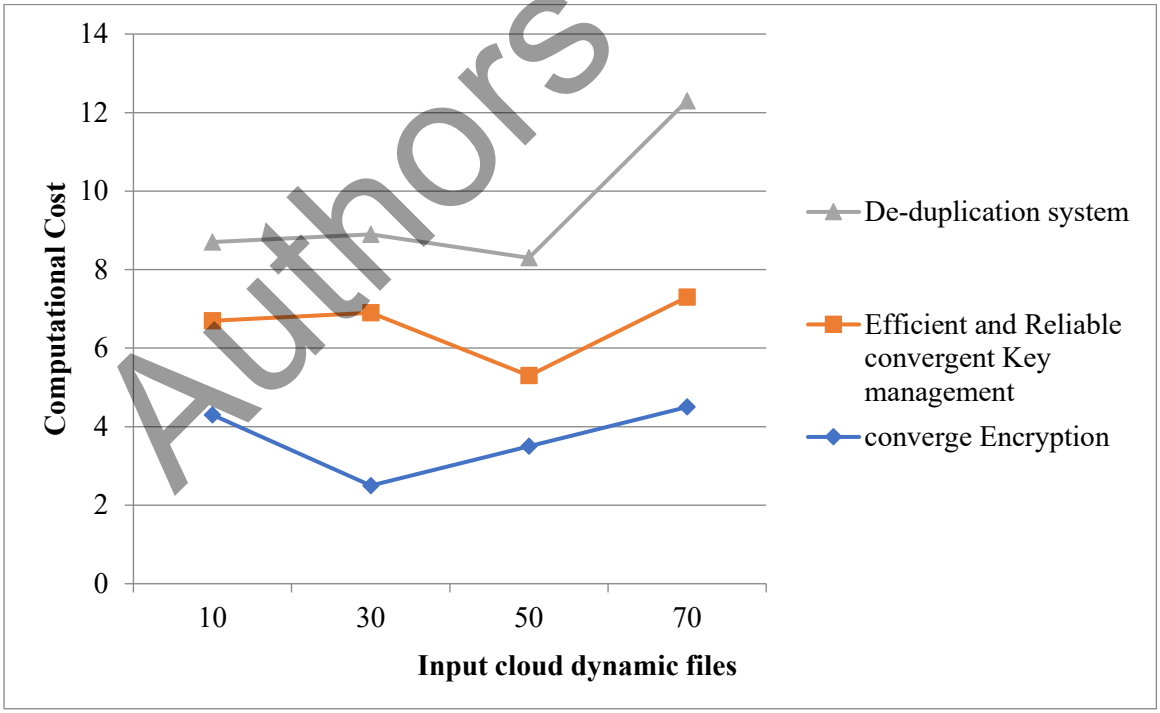


Fig 4.4 Assessment of Processing Computational Cost Performance

Drawing from the aforementioned figures 4.1- 4.4, our suggested methodology yields superior outcomes in terms of data de-duplication concerning encryption and decryption time (see Table 1-3), memory usage for storing multiple files, and

computational expenses for storage and privacy in cloud storage systems [16]. The hash value is already utilized in the file-based de-duplication of process in our method. The hash value is generated using the encrypted file that has been stored. When the data is altered or processed, this hash is compared to the new one to see if there is a difference. The same hash will cause files to be deleted. In the event of a discrepancy, only the freshly inserted section is transferred to storage. It is believed that hashes have one. The hash is updated whenever a piece of the file is changed, giving service providers the ability to update it. These hashes are appropriately indexed and stored. When a file is changed, the new files are computationally hashed and compared to the old ones.

V. CONCLUSION

When used in a blockchain-based data deduplication system, the Dynamic Chunking Algorithm provides a reliable solution that combines efficacy, security, affordability, scalability, and data integrity, making it a viable method for handling data deduplication in a variety of industries. This work proposes a novel data chunking approach to identify incremental information between two similar files. Chunk size instability has little effect in the field of data progressive synchronization because chunks are not kept but are instead used to search for changing data. Resisting byte shifting can help preserve the integrity of unaltered segments, significantly lowering the quantity of altered data discovered. The goal of this approach is to identify incremental data more quickly by increasing byte-shifting susceptibility at the expense of the stability of the chunk sizes in the algorithm's output. Finally, by comparing the interaction, calculation, and storage costs with current methods, we were able to formally demonstrate the effectiveness of the suggested technique. Numerical analysis and experimental results further support the effectiveness of our scheme.

References

[1]. Guo, S., Mao, X., Sun, M., & Wang, S. (2023). Double sliding window chunking algorithm for data deduplication in ocean observation. IEEE Access.

[2]. El Ghazouani, M., Latifa, E. R., & El Khanboubi, Y. (2020). Efficient method based on blockchain ensuring data integrity auditing with deduplication in cloud.

[3]. Sumathi, M. (2021). Secure blockchain based data storage and integrity auditing in cloud. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(9), 159-165.

[4]. Yuan, H., Chen, X., Wang, J., Yuan, J., Yan, H., & Susilo, W. (2020). Blockchain-based public auditing and secure deduplication with fair arbitration. Information Sciences, 541, 409-425.

[5]. Liu, L., Liu, X., & Wan, J. (2022). Design of Updating Encryption Algorithm for Privacy Big Data Based on Consortium Blockchain Technology. Journal of Mathematics, 2022.

[6]. Gnana Jeslin, J., & Mohan Kumar, P. (2022). Decentralized and privacy sensitive data de-duplication framework for convenient big data management in cloud backup systems. Symmetry, 14(7), 1392.

[7]. Zhou, B., Zhang, S., Zhang, Y., & Tan, J. (2015). A bit string content aware chunking strategy for reduced CPU energy on cloud storage. Journal of Electrical and Computer Engineering, 2015, 43-43.

[8]. Zhang, C., Qi, D., Li, W., & Guo, J. (2020). Function of content defined chunking algorithms in incremental synchronization. IEEE Access, 8, 5316-5330.

[9]. Prajapati, P., & Shah, P. (2022). A review on secure data deduplication: Cloud storage security issue. Journal of King Saud University-Computer and Information Sciences, 34(7), 3996-4007.

[10]. Ko, Y. W., Jung, H. M., Lee, W. Y., Kim, M. J., & Yoo, C. (2013). Stride static chunking algorithm for deduplication system. IEICE transactions on information and systems, 96(7), 1544-1547.

[11]. Rao, K. P. (2021). Efficient and Reliable Secure Cloud Storage Schema of Block chain for Data De-duplication in Cloud. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(9), 1547-1556.

[12]. Nisha, T. R., Abirami, S., & Manohar, E. (2016). Experimental study on chunking algorithms of data deduplication system on large scale data. In Proceedings of the International Conference on Soft Computing Systems: ICSCS 2015, Volume 2 (pp. 91-98). Springer India.

[13]. Bo, C., Li, Z. F., & Can, W. (2013). Research on chunking algorithms of data de-duplication. In Proceedings of the 2012 International Conference on Communication, Electronics and Automation Engineering (pp. 1019-1025). Springer Berlin Heidelberg.

[14]. El Khanboubi, Y., Hanoune, M., & El Ghazouani, M. (2021). A new data deletion scheme for a blockchain-based de-duplication system in the cloud. Int. J. Commun. Netw. Inf. Secur, 13, 331-339.

[15]. Rao, K. P. (2021). Efficient and Reliable Secure Cloud Storage Schema of Block chain for Data De-duplication in Cloud. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(9), 1547-1556.

[16]. Paul, S. P., & Vetrithangam, D. (2023, March). A Scientometric Study of Research Development on Cloud Computing-Based Data Management Technique. In Doctoral Symposium on Computational Intelligence (pp. 617-625). Singapore: Springer Nature Singapore.