Journal of Current Science and Technology, September - December 2024 Vol. 14 No. 3, Article 69

ISSN 2630-0656 (Online)

Cite this article: Chansaeng, T., & Khemapatapan, C. (2024). Enhancing multi-cloud storage: a hybrid framework using interleaving and LT codes. *Journal of Current Science and Technology*, *14*(3), Article 69. https://doi.org/10.59796/jcst.V14N2.2024.69

Journal of Current Science and Technology

Journal homepage: https://jcst.rsu.ac.th

Enhancing Multi-Cloud Storage: A Hybrid Framework Using Interleaving and LT Codes

Tanagrit Chansaeng, and Chaiyaporn Khemapatapan*

[1]Department of Computer Engineering, Dhurakij Pundit University, Bangkok 10210, Thailand

*Corresponding Author; E-mail: chaiyaporn@dpu.ac.th

Received 19 December, 2023; Revised 6 February, 2024; Accepted 9 April, 2024
Published online 1 September, 2024

Abstract

This article presents a novel framework for enhancing the integrity and availability of data stored on multi-cloud storage systems. The proposed technique leverages a combination of interleaving and Luby transform (LT) codes, specifically employing either ideal or robust soliton distribution during the LT encoding stage. The proposed technique is rigorously designed, incorporating data preparation facilitated by a cloud broker and culminating in the measurement of key performance indicators (KPIs). A central component of the framework is data processing, which involves interleaving followed by the LT encoding with soliton distribution selection. The performance of the proposed technique is evaluated through data integrity, data availability, and others. The results demonstrate that using the robust soliton distribution offers a pragmatic balance between computational efficiency and robust data integrity, particularly crucial in dynamic multi-cloud environments. Additionally, the proposed technique achieves a high level of data integrity of more than 0.97 and strong resilience with average data availability of 98%. However, further optimization is necessary to address storage overhead, retrieval time, and computational overhead. Despite these considerations, the framework remains significant, especially within the context of confidentiality, integrity, and availability for multi-cloud storage protection. The proposed technique underscores the potential of integrating interleaving and LT codes to revolutionize data storage and access within cloud environments, paving the way for future innovations in the field.

Keywords**:** *data integrity; data availability; multi-cloud storage; interleaving data; Luby transform code; LT code*

1. Introduction

Multi-cloud storage has emerged as a pivotal technology, addressing the growing demands for data storage, accessibility, and security (Jagruthi et al., 2022). The advent of multi-cloud environments represents a significant shift from traditional, monolithic storage solutions to a more distributed, dynamic, and scalable approach. Generally, multi-cloud storage refers to the use of multiple cloud services from different providers to store, manage, and process data. This approach offers several advantages over single-cloud solutions (Imran et al., 2021). Firstly, it mitigates the risks associated with vendor lock-in, providing flexibility and freedom to choose the best services based on cost, performance, and features (Ahmed et al., 2022). Secondly, multi-cloud environments enhance data availability and disaster recovery capabilities (Gu et al., 2014). By distributing data across multiple clouds, organizations can ensure continuous access to their data, even in a service outage or failure in one cloud (Song et al., 2023). Another critical aspect of multi-cloud storage is its contribution to data security and privacy (Antu et al., 2022). With the increasing concerns over data

breaches and cyber threats, multi-cloud storage provides an added layer of protection. Distributing data across various clouds can prevent total data loss or compromise, as the risk is dispersed and not concentrated in a single location. Moreover, multi-cloud storage enables compliance with various regional data protection regulations by allowing data to be stored in specific geographic locations as required by law (George, & Pramila, 2021).

However, multi-cloud storage also presents unique challenges data integrity and availability. While data integrity is crucial for maintaining trust and accuracy in business operations, availability ensures that data is accessible when needed, contributing to an organization's operational efficiency (Izham Jaya et al., 2017). Ensuring data remains unaltered and consistently accessible across different cloud platforms is a complex task (Abdalla, & Varol, 2019). As such, multi-cloud architectures are a testament to the relentless pursuit of enhanced security and operational resilience. Some studies in this field have critically examined the intricacies of deploying and managing disparate cloud services, forging a path toward more robust data protection mechanisms. These research efforts have introduced advanced cryptographic techniques, novel access control models, and sophisticated risk assessment methodologies.

This research focuses on addressing these challenges, particularly the risks of data corruption and unauthorized alterations across different cloud platforms (Witanto et al., 2023). To address these challenges, combining innovative solutions involving advanced coding techniques, such as LT codes and data interleaving methods, are gaining prominence in enhancing data integrity and availability in multi-cloud environments. By enhancing data integrity and availability, the proposed technique seeks to address the core challenges of multi-cloud storage, paving the way for more resilient, efficient, and secure data management practices in the cloud computing era.

1.1 Current Solutions and Limitations

The quest for optimal solutions to ensure data integrity and availability has led to the development of various approaches in multi-cloud storage. This section provides an overview of these existing solutions, intensively analysis of their strengths and identification of the limitations that have necessitated exploring more innovative approaches, such as the hybrid use of Interleave and LT codes.

Several solutions have been proposed to address multi-cloud storage issues. Early solutions primarily revolved around redundancy techniques, such as mirroring and replication across multiple cloud servers, offering essential data availability and integrity. However, they fail in term of efficiency and scalability (Shakarami et al., 2021). Replication, in particular, leads to increased storage costs and complexity in data synchronization, making them less suitable for large-scale cloud environments (Sabaghian et al., 2023). In addition, data deduplication techniques have been employed to reduce storage space by eliminating redundant copies of data. However, they often do not address the core issues of data availability and integrity in multi-cloud environments which are vulnerable to various security threats (Prajapati, & Shah, 2022). While different cryptographic techniques were proposed to address these concerns, they mostly ensure data confidentiality but do not inherently guarantee data availability and integrity. Furthermore, they often impose additional computational overhead and complexity in crucial data management (Por et al., 2023).

There are also several coding solutions introduced to address multi-cloud storage issues. For example, erasure coding emerged as a more storage-efficient alternative to replication, especially in distributed storage systems (Li, & Li, 2013). By dividing data into fragments, encoding them, and storing them across various nodes, erasure coding ensures data recoverability even when some fragments are lost. However, the computational overhead for encoding and decoding, coupled with the increased latency in data retrieval, poses significant challenges, particularly in dynamic cloud environments (Xiao et al., 2020). Furthermore, network coding approaches have been proposed to improve data transfer efficiency in cloud storage offering advantages in data retrieval and repair operations. These solutions, however, introduce implementation complexities and require considerable computational resources, making them less practical for some cloud storage applications (Chen et al., 2014).

Recently, several cloud service providers have developed proprietary solutions optimized for their specific cloud environments. However, they often lead to vendor login and lack the flexibility needed for interoperability across different cloud platforms (Böhm et al., 2019). While hybrid models attempt to combine the benefits of private and public clouds,

offering more effective control and security, they often struggle with complex data management and integration challenges, leading to inefficiencies in data operation and maintenance (Park, & Song, 2013). In addition, vendor login issues can introduce several security vulnerabilities to multi-cloud storage systems. For instance, data privacy (Gupta et al., 2021; Svantesson, 2016; Wylde et al., 2022), breach risks (Cremer et al., 2022), identity management, API security (Crumpler, & Lewis, 2020; Lee et al., 2020), and the complexities of security management. As such, the complexities of multi-cloud environments necessitate a nuanced approach to security, addressing varied challenges.

Table 1 presents a comparative analysis of the existing solutions discussed earlier, encompassing a spectrum ranging from traditional redundancy to hybrid cloud models. These solutions exhibit varying degrees of success when assessed against a set of performance criteria, including efficiency, scalability, reliability, computational overhead, security, data integrity, data availability, and data confidentiality. Notably, the proposed framework demonstrates a high level of achievement across all aspects, except for computational overhead, where it achieves a low degree.

2. Objectives

The objective of this research is to enhance data integrity and availability within a multi-cloud storage environment by developing and implementing a novel framework that integrates data manipulation with data interleaving and LT codes. This framework aims to optimize data management processes and will be rigorously evaluated based on six key performance metrics: data integrity (DI), data availability (DA), storage overhead (SO), retrieval time (RT), computational overhead (CO), and overall effectiveness (OE). The comprehensive assessment of these metrics will provide a detailed understanding of the framework's impact on the efficiency and reliability of multi-cloud storage systems.

3. Materials and methods

As this research aims to achieve two primary objectives: enhancing multi-cloud storage and robust data security, this research leverages a multi-faceted methodological approach focused on security control and management within a multi-cloud environment. The research methodology unfolds in five distinct steps: data preparation, processing, storage, retrieval and reassembly, and retrieval with validation. Notably, data processing encompasses four sub-steps, including interleaving, followed by a combination of LT encoding with either Ideal soliton distribution or robust soliton distribution. Finally, the project culminates in the measurement of DI, DA and other key performance indicators to assess the effectiveness of the proposed approach.

Table 1 A comparison of existing solutions and their limitations with the proposed framework

	Criteria							
Method	**Efficiency**	**Scalability**	**Reliability**	**Computational Overhead**	**Security**	**Data Integrity**	**Data Availability**	**Data Confidentiality**
Traditional Redundancy	Low	Poor	Moderate	Low	Moderate	Low	Moderate	Moderate
Data Deduplication	High	Good	Low	Moderate	Low	Moderate	Moderate	Low
Cryptographic Techniques	Low	Moderate	Low	High	High	Moderate	Low	High
Erasure Coding	High	Good	High	High	Moderate	High	High	Moderate
Network Coding-Based	High	Moderate	High	High	Moderate	High	High	Moderate
Cloud-Specific Proprietary	Varies	Moderate to High	Varies	Varies	High	High	Moderate	High
Hybrid Cloud Models	Moderate	Good	Moderate	High	High	High	High	High
Proposed Framework	High	High	High	Low	Moderate to High	High	High	High

This research employs a multi-faceted methodological approach to achieve the dual objectives of enhancing multi-cloud storage and ensuring robust data security. The methodology is structured into five distinct steps: data preparation, data processing, data storage, data retrieval and reassembly, and retrieval with validation. Data processing includes four sub-steps: interleaving, followed by LT encoding using either Ideal Soliton Distribution or Robust Soliton Distribution. The processed data is then stored across multiple cloud environments to leverage the benefits of a multi-cloud setup. Upon retrieval, the data is reassembled to its original form, ensuring data integrity and availability, followed by validation to confirm these attributes. The effectiveness of the proposed approach will be measured through key performance indicators, including data integrity (DI), data availability (DA), and other relevant metrics, providing comprehensive insights into the efficiency and reliability of the multi-cloud storage and security framework.

3.1 Data Preparation

In a multi-cloud storage system, the cloud broker plays a pivotal role as the intermediary between the user and diverse cloud storage resources. The cloud broker is responsible for managing data processing and distribution across multiple cloud providers. Initially, the user uploads the original data to the cloud broker. To enhance security and data resilience, the cloud broker fragments the original data into 'k' parts. These fragments are then distributed across various cloud storage services. This dispersion strategy mitigates the risks associated with storing data in a single location, thereby enhancing the overall security and robustness of the multi-cloud storage system.

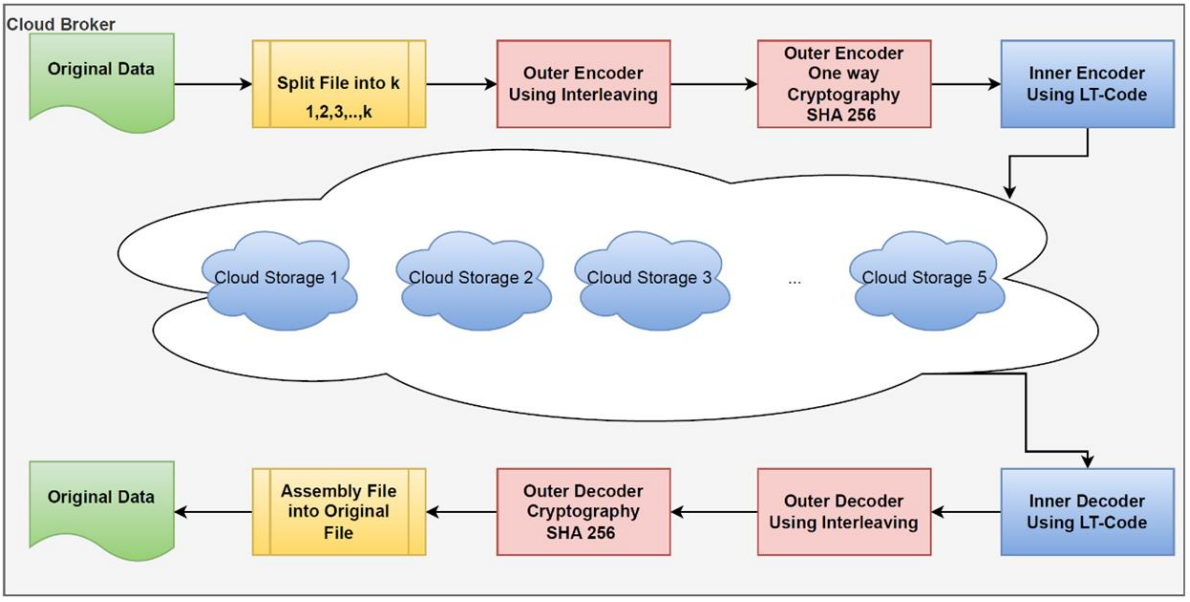

Figure 1 Outer encoder and decoder with interleave scheme

3.2 Data Processing

After fragmenting the original data into 'k' parts, the method employs a multi-step process to enhance data security, integrity, and recoverability. First, the split files are encoded using an interleaving method, which distributes data pieces across various storage units in a non-sequential order, enhancing data recoverability in case of partial data loss or corruption. Next, a cryptographic hash function, specifically SHA-256, is applied to the interleaved data to ensure data integrity and security by transforming the data into an irreversible form without the proper decoding mechanism. Following this, LT (Luby Transform) codes, a type of erasure-correcting code, are used as the inner encoder to enable data recovery even if some parts are missing, which is particularly useful in multi-platform data distribution scenarios. The data processing involves four essential operations: interleaving, LT encoding with ideal soliton distribution, robust soliton distribution, and LT

encoding on interleaves. This comprehensive approach aims to create a robust, secure, and recoverable data storage system in a multi-cloud environment, addressing potential vulnerabilities in data storage and transmission to ensure high levels of data integrity and availability even in the face of partial data loss or corruption.

3.2.1 Interleaving

Interleaving is a technique used in digital systems to enhance data robustness against errors, especially in error-prone environments. It involves reordering data units (e.g., bits, bytes, symbols) before transmission or storage and reversing this order upon reception or retrieval, crucial for maintaining data integrity in digital communications and storage systems (Forney, 1965). Let F represent the original data file partitioned into k blocks $F=\{B_1,B_2,\ldots,B_k\}$. The interleaving function *Intlv* is applied to each block B_i, resulting in an interleaved block $B_i^{'}$. If $B_i=\{b_{i1},b_{i2},\ldots,b_{i3}\}$ where b_{ij} is the *j-th* byte of the *i*-th block, the interleaving of B_i can be represented as: $B_i^{'}=IntlvB_i=\{b_{i\pi(1)},b_{i\pi(2)},\ldots,b_{i\pi(n)}\}$ where π is a permutation function applied to the indices of the bytes within the block.

3.2.2 LT Codes and Ideal soliton Distribution

LT codes, which are rateless erasure code, are designed for reliable data transmission over lossy channels, such as the Internet. They differ from traditional error correction codes by being flexible and efficient, making them well-suited for distributed storage and streaming applications. Key features include their rateless nature, which allows for the generation of an infinite number of encoded symbols from a set amount of source data, optimizing for erasure correction rather than error correction, and employing probabilistic encoding for resilience, and facilitating efficient decoding with minimal overhead (Luby, 2002). In this case, the Ideal soliton distribution is essential in the LT code encoding process, guiding the selection of degrees for encoded symbols. In LT codes, encoded symbols are formed by randomly merging a specific number of original data symbols or 'degree'. The Ideal soliton distribution determines the distribution of these degrees, optimizing the encoding process for efficient data transmission and recovery in environments prone to data loss. The distribution formulas are defined below. The probability distribution $\rho(i)$ for a degree i is given by:

$$\rho(1)=\frac{1}{k} \quad (1)$$

$$\rho(i)=\frac{1}{i*(i-1)};\ where\ i=2,3,4,\ldots,k \quad (2)$$

$$\sum_i \rho(i)=1;\ where\ i=1,2,\ldots,k. \quad (3)$$

3.2.3 LT Codes and Robust Soliton Distribution

The robust soliton distribution is a crucial enhancement of the Ideal soliton distribution, specifically designed for use in LT codes. This distribution modifies the Ideal soliton distribution to improve the practical performance of LT codes, particularly in the decoding process. The robustness it introduces is especially valuable in ensuring successful decoding with a high probability, even when some encoded symbols are lost or corrupted during transmission. The operation of the robust soliton distribution is defined below.

1. The robust soliton distribution adds an additional layer to the probability distribution. This distribution is overlaid on top of the Ideal soliton distribution to increase the likelihood of having more encoded symbols with smaller degrees.

2. The τ distribution (Tau distribution) is calculated based on δ (the failure probability of the decoding process), R (the size of the original data set and δ), and k (the number of original data symbols).

$$R_{SD}=C*ln(k/\delta)\sqrt{k} \quad (4)$$

3. The robust soliton distribution is formed by combining the Ideal soliton Distribution (ρ) and the τ distribution, then normalizing to ensure that the probabilities sum up to 1; where β is a normalization factor to ensure the sum of the distribution equals 1. The combined distribution is given by:

$$\mu(i)=\frac{\rho(i)+\tau(i)}{\beta} \quad (5)$$

4. The distribution ensures that the low-degree symbols are enough for the decoding process to start effectively (thanks to the Ideal soliton part). Moreover, intermediate-degree symbols to maintain the decoding momentum (contributed by the τ part) are concerned to avoid stalls and ensure robustness.

$$\tau(i)=\begin{cases}\frac{R}{ik}\ ,\ i=1,\ldots,\left(\frac{k}{R}\right)-1\\ \frac{R*ln\frac{R}{\delta}}{k}\ ,\ i=\frac{k}{R}\\ 0\ ,\ i>\frac{k}{R}\end{cases} \quad (6)$$

where $\sum_{i=1}^{k} \rho(i) + \tau(i)$. (7)

3.2.4 LT Codes and Interleaves

The LT codes encoding function $LTEnc$ operates on the interleaved blocks B_i'. Let $\Omega(x)$ be the degree distribution used by LT codes. For each encoded symbol s, a degree d is chosen from $\Omega(x)$, and d interleaved blocks are randomly combined to form s. The encoding of a symbol can be mathematically represented as: $s=LTEncB_i'=\bigoplus_{j=1}^{d} B'_{\pi(j)}$ where $\oplus$ denotes the bitwise XOR operation, and $\pi(j)$ is an index chosen according to a pseudo-random number generator as illustrated in Figure 2.

3.3 Data Storage

These entities represent the various cloud storage services that act as repositories for the encoded data fragments. The adoption of a multi-cloud storage architecture, where data is dispersed across CS_1 (Cloud Storage 1) to CS_n (Cloud Storage n), is a well-established strategy for achieving data redundancy. This approach enhances data availability by ensuring that the data remains accessible even if one or more cloud storage providers experience an outage or service disruption. Furthermore, geographically distributed cloud storage backends can offer additional benefits, such as improved latency for geographically dispersed users and enhanced compliance with data residency regulations in specific regions. Using multiple storage clouds is a common strategy to achieve redundancy and improve data availability. For the cloud storage distribution model, let $CS = \{CS_1, CS_2, \ldots, CS_n\}$ denote the set of cloud storage nodes. The distribution function D maps each encoded symbol s to one or more cloud storage nodes: $D(s) \rightarrow \{C_{\alpha 1}, C_{\alpha 2}, \ldots, C_{\alpha 3}\}$ where C_α is the storage node selected for the symbol s, and m is the number of copies of the symbol distributed across the cloud storages.

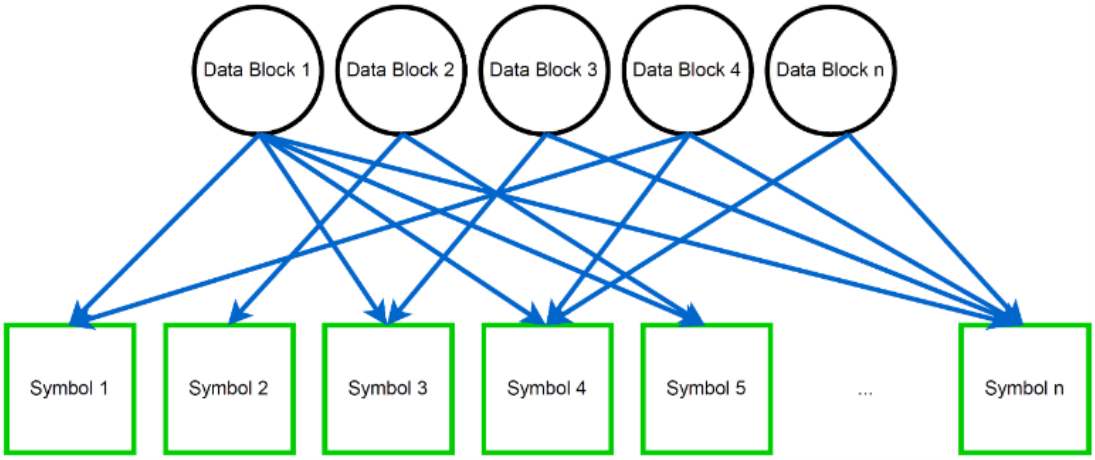

Figure 2 Symbol creation from LT code

3.4 Data Retrieval and Reassembly

Retrieving data stored within a multi-cloud storage system necessitates the reversal of the fragmentation and encoding processes. When the data needs to be retrieved, the parts must be assembled back into the original file. To achieve this, the cloud broker initiates a coordinated retrieval operation across CS_1 to CS_n. The dispersed data fragments are then reassembled in the correct order, leveraging the redundancy introduced during the initial fragmentation stage. This reassembly process ensures the reconstruction of the original file in its entirety, ready for user access or further processing. Notably, the employed erasure-correcting codes play a crucial role during retrieval. These codes enable the system to tolerate a certain level of data loss or corruption within the fragments. Even if a limited number of fragments are unavailable or compromised, the remaining fragments can be mathematically utilized to reconstruct the missing data, ensuring complete file retrieval.

During data retrieval, an objective function O is used to determine the weight or priority for retrieving symbols from each cloud storage. The retrieval function R selects the symbols based on their weights: $R(s) \leftarrow O(C_\alpha)$ The LT codes decoding

function $LTDec$ takes the retrieved symbols to reconstruct the original interleaved blocks: $\{B_1', B_2', \ldots, B_3' = LTDec(\{s_1, s_2, \ldots, s_m\})\}$. After that, the original data file F is reconstructed from the decoded interleaved blocks by reversing the interleaving function $Intlv^{-1}$ and concatenating the blocks in their original order: $F = \cup_{i=1}^{k} Intlv^{-1}(B_i')$.

3.5 Data Retrieval and Validation

The data retrieval process culminates in data validation, ensuring the integrity and accuracy of the retrieved data, as illustrated in Figure 3. This validation phase involves a series of decoding steps that reverse the encoding operations performed during data storage. First, the SHA-256 hash, acting as the outer decoder, is used to verify data integrity. Any alterations to the data during storage or transmission result in a mismatch between the calculated hash and the stored hash value, effectively detecting potential data corruption. Subsequently, the interleaving process is reversed, reordering the data fragments back to their original sequence. Finally, the LT codes, employed as the inner decoder, are utilized to rectify any errors or data losses that might have occurred during storage or transmission. This error correction capability ensures that even if a limited number of fragments are corrupted or unavailable, the remaining fragments can be leveraged to reconstruct the missing data, guaranteeing the complete and accurate retrieval of the original file for user access or further processing. To verify the integrity of the data, a cryptographic hash function H, such as SHA-256, is applied to each block before and after storage. $VerifyIntegrity(B_i') = (H(B_i') == H_i)$ is the hash of the block B_i' before it was stored.

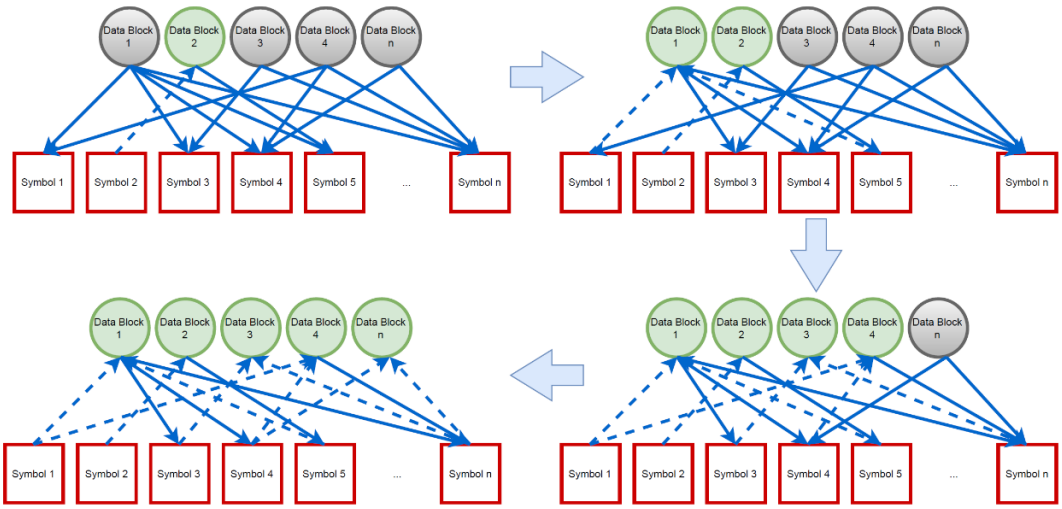

Figure 3 Data Retrieval and Validation

3.6 Measuring DI, DA, and other key performance indicators

To ensure the validity and robustness of the research results, a comprehensive mathematical methodology for performance evaluation has been established. This approach focuses on measuring key performance indicators that are critical to assessing data integrity and availability in the multi-cloud storage system. The study employs six essential metrics: Data Integrity (DI), Data Availability (DA), Storage Overhead (SO), Retrieval Time (RT), Computational Overhead (CO), and Overall Effectiveness (OE). These metrics provide a holistic view of the system's performance, allowing for a thorough assessment of the proposed multi-cloud storage solution.

1. DI: A quantitative assessment of the likelihood that a stored file can be flawlessly reconstructed following multiple retrieval attempts. This metric is operationalized through the following equation:

$$DI = \frac{Number\ of\ correctly\ reconstructed\ blocks}{Total\ number\ of\ blocks} \quad (8)$$

2. DA: A probabilistic metric characterizing the likelihood of successful data access despite potential failures within the underlying cloud storage

infrastructure. The calculation of this metric relies on the following equation:

$$DA = \frac{Number\ of\ successful\ retrievals}{Total\ number\ of\ retrieval\ attempts} \quad (9)$$

3. SO: This metric quantifies the additional storage space necessitated by the encoding process when comparing the two systems under study. We can mathematically represent this metric as follows:

$$SO = \frac{Total\ storage\ used\ after\ encoding}{Original\ data\ size} \quad (10)$$

4. RT: This metric quantifies the temporal expenditure associated with locating and subsequently interpreting stored data. This metric is operationalized through the following equation:

RT= Average time taken for data retrieval and decoding (11)

5. CO: This metric quantifies the additional computational resources necessitated by the encoding and decoding processes employed for data security. The calculation of this metric is governed by the following equation:

CO = Average CPU time for encoding and decoding operations (12)

6. OE: To comprehensively evaluate system performance, an all-encompassing metric, OE will be calculated. This metric will be derived from a weighted formula that incorporates all six performance indicators (DI, DA, SO, RT, CO). Weights (w_i) will be assigned to each metric, reflecting their relative significance in the context of the specific application. The calculation of this metric relies on the following equation:

$$OE = w_1*DI + w_2*DA + w_3*(1-SO) + w_4*(1-RT) + w_5*(1-CO) \quad (13)$$

4. Results

4.1 Degree of distribution for LT Codes

A comparative analysis of ideal soliton and robust soliton degree distributions for LT codes in multi-cloud storage, as shown in Figure 4, revealed that the robust soliton distribution is better suited for practical applications, offering greater resilience to adverse conditions. Though imposing slightly higher overhead, the robust soliton redundancy and uniformity of the robust soliton distribution mitigated the data integrity risks inherent in complex multi-cloud ecosystems. Its recovery capabilities, despite disruptions, align with the principal aims of availability and reliability. Thus, when combined with interleaving techniques, the robust soliton distribution provides a prudent balance between computational efficiency and the paramount priority of robust data integrity assurance in volatile multi-cloud environments.

4.2 Data Integrity

In the 100-iteration simulation of the DI experiment, the results, as demonstrated in Figure 5, indicated a high level of data integrity in a multi-cloud storage framework utilizing interleaving and LT codes, with DI values primarily above 0.97. The consistency in DI metrics suggests reliable performance despite fluctuations attributed to simulated random errors, highlighting the framework's resilience. Even with some data blocks corrupted, the integrity rarely dropped below 0.97, demonstrating its potential suitability for multi-cloud environments where data integrity is critical. Although the high DI indicated effective maintenance of data accuracy, the variations suggest opportunities for further optimization in encoding and decoding algorithms. The experiment provides strong evidence of the framework's effectiveness in maintaining data integrity, pointing towards its reliability for multi-cloud storage applications and indicating areas for future research and system optimization.

4.3 Data Availability

A 100-iteration Python script simulation evaluated DA within a multi-cloud storage framework, as depicted in Figure 6. The system demonstrated strong resilience, maintaining a 98% average DA despite a simulated 10% node failure probability. This high availability highlights the built-in redundancy and fault tolerance vital for uninterrupted access. The integrated interleaving and LT coding techniques significantly bolstered data recovery capabilities. However, a 2% DA failure rate under simulated node failures indicates room for optimizing error correction and distributed data mechanisms to further strengthen multi-cloud storage reliability for mission-critical services. Overall, the positive results affirm the framework's efficacy in ensuring consistent data access while pointing toward future enhancements to mitigate the failure risks inherent to complex multi-cloud ecosystems.

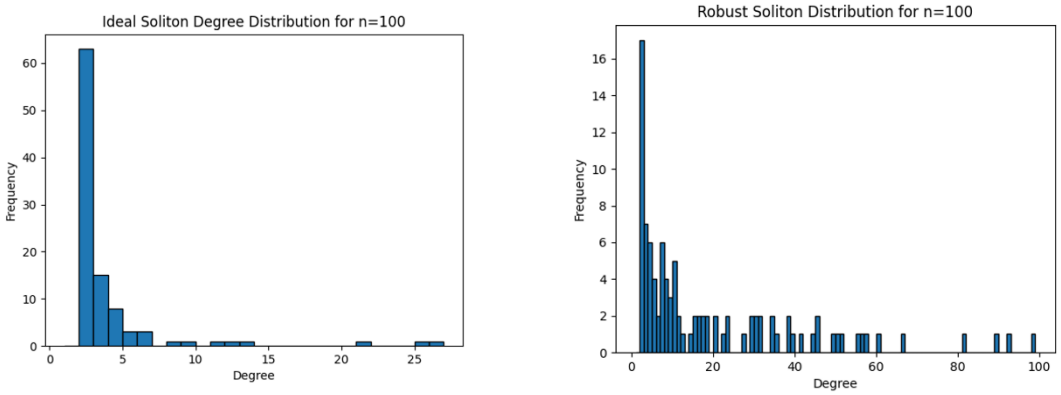

Figure 4 The distinction between ideal soliton and robust soliton degree distribution

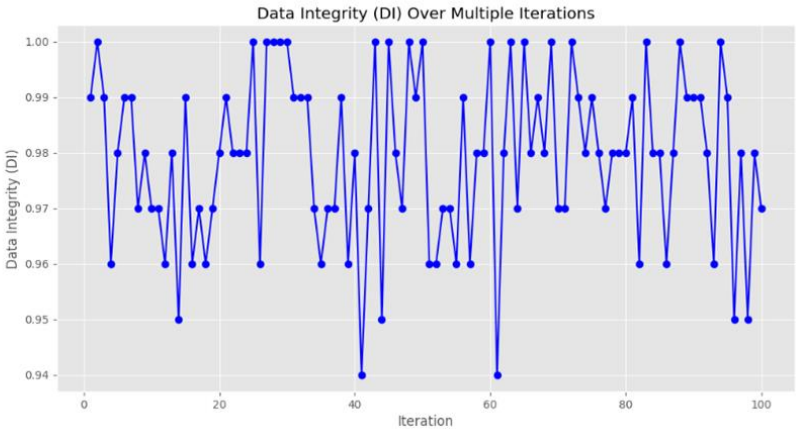

Figure 5 Data Integrity (DI) over multiple iterations

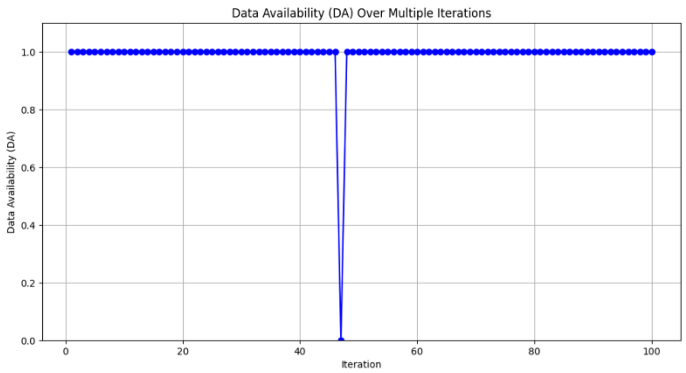

Figure 6 DA with occasional drop in multi-cloud storage.

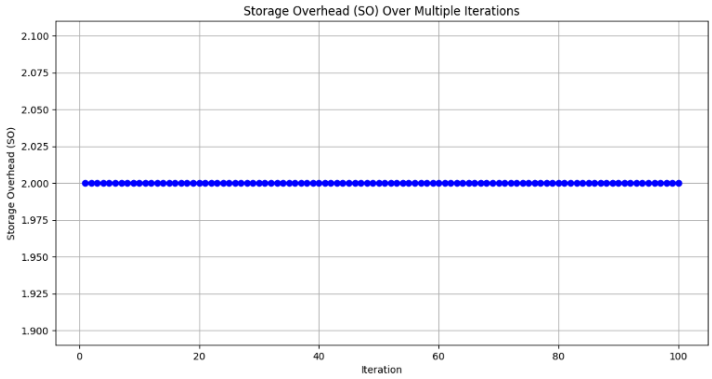

Figure 7 Consistent storage overhead in multi-cloud storage framework simulation

4.4 Storage Overhead

The simulation across 100 iterations was conducted using Python programming to determine the storage overhead as the result shown in Figure 7. It indicates that a consistent doubling of the original data size is reflecting a fixed encoding with overhead factor of 2. This significant increase in storage size underscores a trade-off between enhanced data integrity and availability and increased storage costs and resource utilization, which is particularly relevant for large-scale data applications. It highlights the need to optimize redundancy in relation to storage efficiency and cost, especially for sensitive or critical data. It suggests that strategic adjustments to the encoding process could mitigate overhead impacts while preserving data integrity and availability.

4.5 Retrieval Time

The simulation of RL over 100 iterations in a multi-cloud storage framework using interleaving and LT codes, as illustrated in Figure 8, revealed significant insights into the efficiency of the data retrieval process. The results showed variability in RL, reflecting the real-world influences of network conditions, server load, and data complexity on the time taken to retrieve and decode data. The latency was influenced by both the retrieval time from the cloud and the decoding time, with variations attributable to network speed and algorithm efficiency. This variability indicated that some data retrieval processes were faster than others, potentially impacting applications that require rapid data access. The need for optimization became apparent, particularly in improving cloud architecture and decoding methods to reduce latency. As the system scales, maintaining acceptable latency levels becomes crucial. Benchmarking against other systems could provide context on performance, and implementing real-time monitoring and adaptive techniques might help manage RL dynamically. Overall, the simulation highlighted areas for optimization in the multi-cloud storage framework to ensure fast and consistent data access.

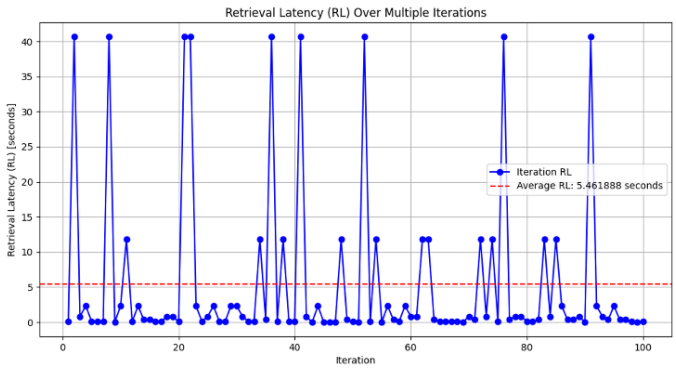

Figure 8 Variability in RL Across Iterations in Multi-Cloud Storage

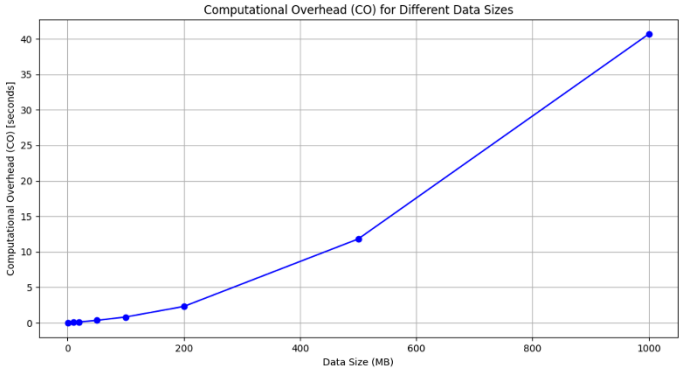

Figure 9 Proportional increase in CO with data size in multi-cloud storage

4.6 Computational Overhead

The simulation and analysis of the CO in a multi-cloud storage framework using interleaving and LT codes with real-world data, as shown in Figure 9, yielded insightful results. It was observed that CO increased with the data size, establishing a direct correlation between data volume and computational resources needed for encoding and decoding. This trend indicated the framework's scalability challenges, especially for larger datasets. The nearly linear relationship between data size and CO suggested a proportional increase in computational workload, underlining the need for efficient resource allocation in cloud storage systems. The increasing CO with larger datasets emphasized the necessity of optimizing encoding and decoding algorithms for efficiency, particularly vital for large-scale applications like big data analytics. These findings pointed towards potential improvements in LT codes and interleaving techniques to reduce CO and highlighted the importance of resource planning and management based on computational demands. Furthermore, the results served as a benchmark for comparing different cloud storage frameworks, aiding in the selection of the most suitable system for specific applications. Overall, the experiment underscored the importance of balancing performance and computational overhead in the storage framework, especially as data volumes scale.

4.7 Overall Effectiveness

A structured approach was employed to experimentally determine the weights *w* sub *i*. for each performance metric in a multi-cloud storage framework, focused on aligning with the CIA triad of real-world applications. The first step involved understanding the relevance of each aspect of the CIA triad in the context of the cloud storage system. Confidentiality, although not directly measured by metrics like DI, DA, SO, RT, or CO, influences the importance placed on DI and DA. DI was emphasized for maintaining data trustworthiness, and DA is critical to ensuring consistent data access.

In the second step, weights were assigned to each metric based on their significance in the framework. For example, a higher weight was given to DI if data integrity was paramount, reflecting the 'Integrity' aspect of the CIA triad. Similarly, DA was weighted more heavily if constant data access is crucial, aligning with 'Availability'. Weights for SO and CO are prioritized if minimizing resource usage is a key concern, and RT is given more importance in systems where quick data access was essential. Additionally, conducting surveys or panels with industry experts and stakeholders helps assess the significance of each CIA aspect, aiding in assigning appropriate weights. After initial assignments, experiments are conducted to observe the impact of these weights on the OE. Based on outcomes, weights are adjusted to better reflect system priorities and performance. For instance, a scenario where data integrity and availability are primary concerns might lead to weights like w_1 (DI Weight) = 0.4 and w_2 (DA Weight) = 0.3, aligning with the focus on Integrity and Availability in the CIA triad. This systematic approach ensures that the OE calculation accurately reflected real-world priorities and the core principles of the CIA triad.

4.8 The experiment with different data sizes

Empirical evaluations of a multi-cloud storage framework employing interleaving and LT codes disclosed significant insights into its operational performance, as demonstrated in Figure 10. The

findings highlighted a decline in encoding and decoding speeds with increased data sizes, attributed to elevated computational demands. Specifically, encoding speeds diminished by 59% when data sizes escalated from 1MB to 1000MB, whereas decoding speeds plummeted by approximately 98.5%. This indicates a potential bottleneck in the decoding phase for large-scale data, which recorded the experimental results with varying data sizes, as shown in Table 2. Time complexity analyses revealed a pronounced sensitivity of decoding times to data sizes, adversely affecting system responsiveness and user experience at larger scales. Despite high initial efficiency, performance inefficiencies at more significant data volumes underscore the necessity for algorithmic enhancements or more robust computational resources to preserve high-level performance. The block and symbol generation illustrated the rateless nature of LT codes for varied data sizes. This demonstrated high efficiency for small datasets, which is advantageous for real-time applications but encountered scalability issues with larger datasets, impacting suitability for big data and real-time processing needs. Future work should focus on optimizing decoding speeds for large datasets, possibly through parallel processing or hardware accelerations, such as GPUs, to maintain high performance across diverse applications in multi-cloud environments.

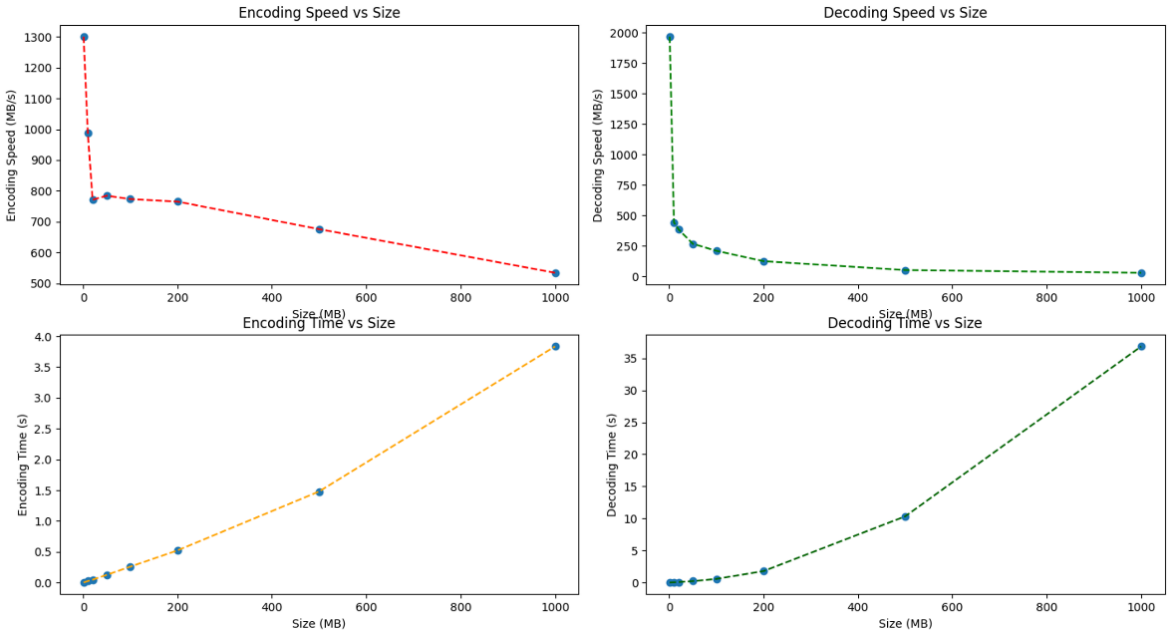

Figure 10 Encoding and decoding performance by data size in a multi-cloud storage framework

Table 2 Data Size vs. Encoding/Decoding Performance in Interleaving and LT Codes Implementation

Size (MB)	NumBlock	Blocks	Symbols	Encoding Times(s)	Encoding Speed (MB/s)	Decoding Times(s)	Decoding Speed (MB/s)
1	15.26	16	32	0.001611948	1300.47	0.0014112	1968.76
10	152.59	160	320	0.025127888	988.04	0.036130905	442.7
20	305.18	320	640	0.047547102	770.92	0.076231956	384.53
50	762.94	765	1530	0.12563014	784.11	0.218972921	267.11
100	1525.88	1600	3200	0.259145975	773.16	0.565399885	209.3
200	3051.76	3200	6400	0.523110867	765.02	1.783993959	124.23
500	7629.39	8000	16000	1.480288029	675.55	10.32928205	51.35
1000	15258.79	16384	32768	3.835714102	533.94	36.84235787	29.06

5. Discussion

This study presents an innovative approach to enhance data integrity and availability in multi-cloud storage through the integration of interleaving and LT codes. The hybrid method significantly improves data protection and error correction, addressing key multi-cloud storage challenges. While introducing computational overheads, the framework demonstrates potential for broad applicability across various cloud storage settings, from individual to enterprise levels. The framework's significance lies in its ability to effectively tackle core multi-cloud storage issues, surpassing existing models in ensuring data integrity and availability. Empirical evidence supports its advantages over traditional approaches. Its potential to redefine industry norms for cloud data management is notable, offering a versatile and reliable solution across diverse sectors.

Future research directions include optimizing the trade-off between computational overhead and system performance, especially considering growing data volume requirements. Additionally, exploring advanced interleaving techniques, LT code variants, and integration with emerging cloud technologies could further enhance multi-cloud environment solutions, addressing challenges such as data migration, cost optimization, and energy efficiency.

This framework represents a pivotal advancement in cloud storage technology, with the potential to shape the future of data storage and management in an evolving cloud computing landscape.

6. Conclusion

The study introduces a framework for enhancing data integrity and availability in multi-cloud storage through interleaving and LT codes, demonstrating significant improvements. Key achievements include augmented data integrity via LT codes error correction and interleaving risk mitigation, superior data availability through LT codes resilience, and efficient storage management leveraging LT codes rateless property for space optimization. These findings underscore the framework's potential in cloud storage environments, highlighting advancements in data restoration accuracy and uninterrupted access amidst cloud infrastructures. While the computational demands of LT codes are notable, the benefits in data integrity and availability justify these costs, suggesting future research on computational efficiency. It confirms the framework's scalability and calls for algorithmic improvements for large-scale data. The framework's practical relevance is affirmed for various cloud storage applications, marking a significant contribution to cloud storage research, particularly in multi-cloud environments. It underscores the potential for future innovations in cloud storage, driven by the integrating of interleaving and LT codes, to revolutionize data storage and access in cloud environments.

7. Acknowledgement

I would like to express my gratitude to Associate Professor Dr. Pita Jarupunphol for proofreading this article.

8. References

Abdalla, P. A., & Varol, A. (2019, June 10-12). *Advantages to Disadvantages of Cloud Computing for Small-Sized Business* [Conference presentation]. 2019 7th International Symposium on Digital Forensics and Security (ISDFS), Barcelos, Portugal. https://doi.org/10.1109/ISDFS.2019.8757549

Ahmed, M. K., Mandebi, J., Saha, S. K., & Bobda, C. (2022). Multi-Tenant Cloud FPGA: A Survey on Security. *arXiv preprint arXiv:2209.11158*. https://doi.org/10.48550/ARXIV.2209.11158

Antu, A. D., Kumar, A., Kelley, R., & Xie, B. (2022). Comparative Analysis of Cloud Storage Options for Diverse Application Requirements. In K. Ye & L.-J. Zhang (Eds.), *Cloud Computing – CLOUD 2021* (Vol. 12989, pp. 75–96). Springer International Publishing. https://doi.org/10.1007/978-3-030-96326-2_6

Böhm, M., Pelz, K., & Schneider, M. (2019). *The Vendor Lock-In Effect of Software: A Case Study about LiMux and Microsoft*. ResearchGate. Retrieved from https://www.researchgate.net/publication/340948942

Chen, H. C. H., Hu, Y., Lee, P. P. C., & Tang, Y. (2014). NCCloud: A Network-Coding-Based Storage System in a Cloud-of-Clouds. *IEEE Transactions on Computers*, *63*(1), 31–44. https://doi.org/10.1109/TC.2013.167

Cremer, F., Sheehan, B., Fortmann, M., Kia, A. N., Mullins, M., Murphy, F., & Materne, S. (2022). Cyber risk and cybersecurity: A systematic review of data availability. *The Geneva Papers on Risk and Insurance - Issues and Practice*, *47*(3), 698–736. https://doi.org/10.1057/s41288-022-00266-6

Crumpler, W. D., & Lewis, J. A. (2020). *Cybersecurity and the Problem of Interoperability*. Center for

Strategic and International Studies (CSIS). Washington, DC.

Forney, G. D. (1965). *Concatenated Codes.* Massachusetts Institute of Technology Cambridge. Massachusetts.

George, S. S., & Pramila, R. S. (2021). A review of different techniques in cloud computing. *Materials Today: Proceedings*, *46*, 8002-8008. https://doi.org/10.1016/j.matpr.2021.02.748

Gu, Y., Wang, D., & Liu, C. (2014). DR-Cloud: Multi-cloud based disaster recovery service. *Tsinghua Science and Technology*, *19*(1), 13-23. https://doi.org/10.1109/TST.2014.6733204

Gupta, R., Saxena, D., & Singh, A. K. (2021). *Data Security and Privacy in Cloud Computing: Concepts and Emerging Trends*. *arXiv preprint arXiv: 2108.09508*. https://doi.org/10.48550/arXiv.2108.09508

Imran, H. A., Latif, U., Ikram, A. A., Ehsan, M., Ikram, A. J., Khan, W. A., & Wazir, S. (2020, November 5-7). *Multi-cloud: a comprehensive review* [Conference presentation]. 2020 IEEE 23rd International Multitopic Conference (INMIC). IEEE. Bahawalpur, Pakistan. https://doi.org/10.36227/techrxiv.13642850

Izham Jaya, M., Sidi, F., Ishak, I., Suriani Affendey, L. I. L. L. Y., & JABAR, M. A. (2017). A Review of Data Quality Research in Achieving High Data Quality within Organization. *Journal of Theoretical & Applied Information Technology*, *95*(12), 2647-2657.

Jagruthi, A., Vikas, K., Nookaraju, G., Teja, K. R., Sanjana, G., & Jabbar, M. A. (2022, October 3-5). *Enhancing Data spillage in Multi-Cloud Storage Services* [Conference presentation]. 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India. https://doi.org/10.1109/ICCCNT54827.2022.9984442

Lee, C. A., Bohn, R. B., & Michel, M. (2020). *The NIST cloud federation reference architecture.* NIST Special Publication 500-332. https://doi.org/10.6028/NIST.SP.500-332

Li, J., & Li, B. (2013). Erasure coding for cloud storage systems: A survey. *Tsinghua Science and Technology*, *18*(3), 259-272. https://doi.org/10.1109/TST.2013.6522585

Luby, M. (2002, November 19). *LT codes* [Conference presentation]. The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings, Vancouver, Canada. https://doi.org/10.1109/SFCS.2002.1181950

Park, G. S., & Song, H. (2013, October 14-16). *A fountain codes-based hybrid P2P and cloud storage system* [Conference presentation]. 2013 International Conference on ICT Convergence (ICTC), Jeju, Korea (South). https://doi.org/10.1109/ICTC.2013.6675310

Por, L. Y., Yang, J., Ku, C. S., & Khan, A. A. (2023). Special Issue on Cryptography and Information Security. *Applied Sciences*, *13*(10), Article 6042. https://doi.org/10.3390/app13106042

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. https://doi.org/10.1016/j.jksuci.2020.10.021

Sabaghian, K., Khamforoosh, K., & Ghaderzadeh, A. (2023). Data Replication and Placement Strategies in Distributed Systems: A State of the Art Survey. *Wireless Personal Communications*, *129*(4), 2419–2453. https://doi.org/10.1007/s11277-023-10240-7

Shakarami, A., Ghobaei-Arani, M., Shahidinejad, A., Masdari, M., & Shakarami, H. (2021). Data replication schemes in cloud computing: A survey. *Cluster Computing*, *24*(3), 2545–2579. https://doi.org/10.1007/s10586-021-03283-7

Song, J., Ma, Y., Zhang, Z., & Liu, P. (2023). An In-Memory Data-Cube Aware Distributed Data Discovery Across Clouds for Remote Sensing Big Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *16*, 4529–4548. https://doi.org/10.1109/JSTARS.2023.3267118

Svantesson, D. (2016). Enforcing Privacy Across Different Jurisdictions. In D. Wright & P. De Hert (Eds.), *Enforcing Privacy: Regulatory, Legal and Technological Approaches* (pp. 195–222). Springer International Publishing. https://doi.org/10.1007/978-3-319-25047-2_9

Witanto, E., Stanley, B., & Lee, S.-G. (2023). Distributed Data Integrity Verification Scheme in Multi-Cloud Environment. *Sensors*, *23*(3), 1623. https://doi.org/10.3390/s23031623

Wylde, V., Rawindaran, N., Lawrence, J., Balasubramanian, R., Prakash, E., Jayal, A., ... & Platts, J. (2022). Cybersecurity, data privacy and blockchain: a review. *SN Computer Science*, *3*(2), Article 127. https://doi.org/10.1007/s42979-022-01020-4

Xiao, Y., Zhou, S., & Zhong, L. (2020). Erasure Coding-Oriented Data Update for Cloud Storage: A Survey. *IEEE Access*, *8*, 227982–227998. https://doi.org/10.1109/ACCESS.2020.3033024