National College of Ireland

Crypto Security Layer for Healthcare Applications and Data Storage in a Multi-Cloud Environment

MSc Research Project
Cloud Computing

Achal Bhangre
Student ID: 22181946

School of Computing
National College of Ireland

Supervisor: Dr. Punit Gupta

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Achal Bhangre
Student ID:	22181946
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr. Punit Gupta
Submission Due Date:	14/12/2022
Project Title:	Crypto Security Layer for Healthcare Applications and Data Storage in a Multi-Cloud Environment
Word Count:	5828
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Achal.Bhangre
Date:	24th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Crypto Security Layer for Healthcare Applications and Data Storage in a Multi-Cloud Environment

Achal Bhangre
22181946

Abstract

There has been a rise in the amount of individuals storing and using data in multi-cloud hosting systems in recent years. Among the benefits it offers are the assurance of data safety, the avoidance of information corruption, and the prevention of unethical issues arising from suppliers. The hybrid approach presented in this project was designed with the goal of enhancing cloud data security and privacy. The hybrid approach makes use of a multi-cloud hosting setup. This hybrid methodology consists of two different parts. (A) an encryption layer that withstands security lapses and permits uninterrupted operation of the application. (b) The confidentiality and reliability of data stored in the cloud are enhanced by the use of encoding and decoding techniques via multi-cloud architecture. The use of encryption techniques improves data storage privacy and trustworthiness without sacrificing efficiency. For a range of medical datasets, approaches including hybrid encryption and multi-level encryption are used and compared with the privacy and security issues related to the hybrid approach. The hybrid approach outperforms existing multi-level encryption techniques like Twofish, ChaCha2-, Serpent, Camellia, and TripleDES in terms of performance. This holds throughout the amount of memory used, the time needed for encryption and decryption, and the overall time needed for authentication. The hybrid technique provided favorable results for the average accuracy measurements, the amount of memory utilized, and the time required for encryption and decryption.

Keywords— Cloud Services, Healthcare Application, Data Security, Encryption, Strategically Store Data

Table of Contents

1 Introduction

The security of cloud computing is receiving a lot of interest, not just from the academic research community but also from several different industries. This results in the development of "everything as a service," which paves the way for information to be accessed via the web. The three main ideas in cloud computing Sajjan and Ghorpade (2019) are platform, infrastructure, and SaaS (software as a service). Security majors with Authentication, privacy, virtualization, secrecy, scaling large quantities of computation of information, and access control are only some of the major security concerns that multi-cloud computing (MCC) brings.

Multi-cloud storage (MCS) separates the user's data block into file objects and distributes them across CSPs, enabling a few CPSs to retrieve the whole file chunk, increasing security, secrecy, and access. Private cloud storage is normally managed as a single entity. MCS integrates many cloud providers' storage solutions. Organizations gain flexibility, security, data durability, compliance, and cost savings. Increasing multi-cloud security is crucial. IoMT, telemedicine, and e-healthcare can deliver medical images over the cloud using 5G technology. The main challenges in cloud-based medical picture transmission are digital signature (DS), integrity, data access control (DAC), security, and confidentiality (SC).

The Components of a Secure Healthcare Application Digitization in healthcare implies establishing enormous online medical records that engage patients and the whole healthcare industry. In addition, healthcare providers are integrating IoT and smart devices into healthcare apps to facilitate real-time patient monitoring and cut down on unnecessary hospitalizations for checkups. Electronic health records (EHRs) are a crucial resource for any healthcare organization. In the present healthcare apps, one person is in charge of keeping all of the electronic health records up-to-date. As a result, the CIA triad of electronic health records (EHRs) is jeopardized by a wide variety of security vulnerabilities in traditional healthcare applications. Therefore, severe problems may arise throughout the treatment due to the altered medical records. There are other hazards of internal and external threats, such as unlawful access and information exposure.

There is an immediate need to properly understand and solve the security and privacy issues of Health applications from the software system's life cycle Figure 1 perspective since these concerns have emerged as the most challenging aspects for healthcare information systems.

In recent years, a new paradigm for cloud storage services has emerged: multi-cloud storage. Users are not limited to a single cloud for archival purposes; several clouds can be used. Leakage risks are mitigated thanks to this framework's design. A multi-cloud system's storage Pushpa (2020) strategy can be dynamically selected based on the current conditions of all cloud providers. In the event of a failure at one cloud provider, the system will automatically switch to a backup cloud.

Massive data sets are increasingly valued as community assets. In several fields, the amount of data is measured in terabytes and even petabytes. Cloud data centres are commonly used to house this kind of massive data. Data replication is the standard

method for dispersed management of huge datasets at present. Replicating data makes it possible to increase data availability while decreasing access latency. A basic and widely used way to hide and protect secret information is through cryptography. Using encryption methods, cryptography turns raw data into cipher text to keep data safe while it is being sent over a network or stored. These days, cryptography Weir et al. (2023) is used for different reasons, like preserving the privacy and security of data.

This study proposal presents a tiered structure for delivering security as a service in cloud Weir et al. (2023) environments, concerning application security during file transmission. Based on the size of the file, this framework's security service offers encryption protection in diverse medical datasets.

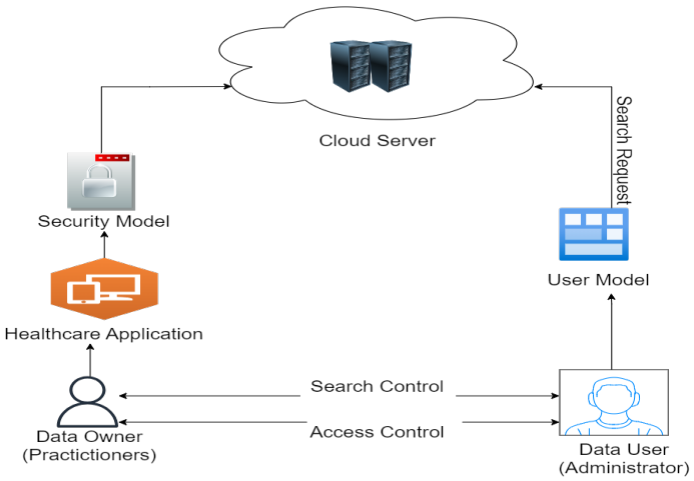

Figure 1: Healthcare applications model

1.1 Research Question

Q 1. How can we improve the quality of healthcare applications used by users to secure and store which will handle the parameters of network latency in the multi-cloud environment?

Q 2. What methodology is used to achieve two public cloud providers and data security and privacy measures for healthcare data by improving network latency in multi-cloud areas?

1.2 Paper Structure

The methodology is further elaborated upon in the next sections of the study.
Section 2, we talk about the literature review, which brought up problems with healthcare data protection and different ways to fix them.
Section 3 is Methodology part, with subsections on Process flow.
Section 4 is about Design Specification. It is split into two key parts: FlowChart Diagram and Performance Indicator.

Section 5 shows the execution phase, which includes the dataset, model architecture, performance calculations, and the results that were found.
Section 6 talks about the evaluation phase, which is divided into four parts: the data encryption approach, the model of the cloud resource, the dataset with different encryption models, and the discussion.
Section 7 is completion and outlines possible future work .

2 Related Work

The related work compiled in this paper consists of Healthcare application Security, Multi-Cloud, and encryption model aspects for data security. Each part is described with details and the latest to oldest paper hierarchy.

2.1 Healthcare application Security

Papers on healthcare sector risk modelling, biomedical application security, and steganography technique are included in Healthcare Application Security.

The authors' Weir et al. (2023) primary focus was on the importance of security and privacy for health apps and devices. Their primary objective is to design and build an assisted workshop that developers can use to conduct risk assessments. This workshop will follow a healthcare sector risk model and consist of an organized set of exercises. They used a workshop-based approach, which may have been time-consuming but yielded valuable insights into the project's design.

Authors Aski et al. (2019) concern with Wireless Computing Technologies (WCT) are inherently vulnerable, securing biomedical applications offered by the Internet of Things (IoT) has long been an important problem. In a protected resource-accessing setting, the suggested model helps with the construction of a generic security framework for healthcare applications that is based on two-way authentication. Readers can gain a better grasp of how to deal with security breaches by delving into the topic of authentication and verification models. Using the authentication and key generation phase shows how to implement a multilayered security architecture that is centred around two-way authentication. This paper's design methodology takes into account the benefit of key mechanisms.

Security and integrity of medical data have emerged as major concerns in the healthcare industry's use of the Internet of Things (IoT), prompting the authors Elhoseny et al. (2018) to suggest a steganography approach using a hybrid encryption scheme. Rivest, Shamir, and Adleman algorithms are used in conjunction with the Advanced Encryption Standard to construct the hybrid encryption schema. The stego-picture was compared to the original medical cover image. Along with encryption testing, the suggested model demonstrated its capacity to conceal sensitive patient data, outperforming state-of-the-art approaches.

2.2 Multi-Cloud

The papers in the Multi-Cloud part include topics such as symmetric encryption techniques, resource accessibility, and joint optimization models.

Writers Alam et al. (2021) inspired by One of the most basic and complicated research problems in the cloud is how to allocate and manage resources while keeping service quality in mind. A straightforward and fast solution is to employ a genetic technique. The recommended joint optimization model considers both the historical performance of the CSPs and the present constraints on their resources. Modifying the model's constituent parts allows it to adapt to a wide range of business requirements. Their goal is to bridge the gap between various cloud service providers by offering a safe and dependable methodology for managing trust. With the default security considerations in cloud service providers, these factors are introduced in this methodology.

By working together in a multi-cloud environment Anwarbasha et al. (2021), service providers improve resource accessibility and offer dynamic operations that can execute simultaneously. For data storage in a multi cloud setting, this article introduces DA-ICP, a Dynamic Level Based Integrity Checking Protocol. To ensure that the server has the original data without downloading it, the suggested solution provides the Provable Data Possession (PDP) approach. This allows users to outsource data to untrusted multi-clouds. To further increase the efficiency of the presented model, we take into account the importance of data integrity and use linear congruential generators for random sequence creation. More trustworthiness is provided by this paper because it takes the supplementary aspects into account.

The authors Dr Ramalingam Sugumar (2018) express concern about the data's security and fitness. Customers frequently entrust cloud storage providers with sensitive data, however, there is a risk that these providers pose security risks. The primary concern with cloud storage is data security. To ensure the safety of data stored in the cloud, this research suggests a symmetric encryption technique. The generated key is the principal means of user authentication. The user guarantees that the data is saved exclusively on protected storage and cannot be read by administrators or intruders by implementing this encryption scheme. The use of a singular cloud application to manage a wide range of data types could appear to be a restriction. To address storage concerns, this proposed design incorporates the Multi-cloud application.

2.3 Encryption model aspects for the data security

In this part, you will find articles about symmetric cryptographic algorithms like AES, anonymization techniques like Triple DES, and optimal circuit-based encryption sub-block implementations.

Many data-sensitive applications face serious issues when it comes to the security and privacy of IoT devices. In this paper Rashidi (2021), authors offer hardware implementations of the Camellia block cipher that are both versatile and designed for high-throughput Internet of Things applications. Implementing encryption sub-blocks based on optimized circuits is the goal of the suggested architectures. To implement the encryption process

and generate intermediate key values at two different times, the structures that Camellia suggests are created and shared. For this project's implementation phase, the best performance was found when using the proposed 128-bit key structure of the Camellia encryption.

Author's Devi and Chamundeeswari (2020) Concerning the healthcare industry and to overcome security difficulties in the current method, the authors offer a security and privacy-preserving framework for big data. The main focus is on incorporating anonymization techniques, such as Triple DES, for security purposes. To prevent these kinds of assaults, Triple DES provides a straightforward method of raising DES's key size, which doesn't need the development of a brand-new block cipher algorithm. Built with big data healthcare security in mind. On the JAVA with Cloud Sim platform, the suggested safe big data storage is implemented. Part of a multi-cloud setup is encrypted using the Triple DES method in this implementation part.

Virtualization, multi-user support, scalability, and a host of other features are all part of Multi-cloud which is considered by Author's Lee et al. (2018). Cloud storage allows customers to access and organize their data on an as-needed basis, which is a crucial feature it offers. An unauthorized individual can gain access to this data via the virtual machines. Users face a significant challenge due to this insecurity. Thus, cloud computing data security is an important issue. As of right now, the most widely used symmetric cryptographic algorithm is AES. Because of its speed, adaptability, scalability, and ease of implementation, AES encryption is the preferred approach. The use of 128-, 192-, or 256-bit keys gives the AES algorithm a very high degree of security. Taking these factors into account is ideal when choosing a cloud service that uses AES as its default encryption method.

3 Methodology

3.1 Process Flow

The study question and goal was "How can we improve the quality of healthcare applications that users use to store and protect data and that can handle network latency in a multi-cloud environment?" How do you improve network delay in places with more than one cloud so that you can use two public clouds and take steps to protect healthcare data's security and privacy?

In the healthcare industry, data collection refers to the process of gathering, evaluating, and applying information for patient records and resources. Because of technology, patient data can be accessed instantly across the whole healthcare system. Additionally, by working together, healthcare systems can increase the accuracy of the data they gather. Usage of quantitative data in various file formats. Quantitative data may be found in text, pictures, multimedia, numerical reports, etc. One may get secondary data from sources that are about to become widely used, such as wearables, mobile applications, electronic health records, genome sequencing, and the Internet. The ability to upload

multiple files will be available in the program. Stated differently, a solitary user will possess several alternatives for submitting files. The information covers the following traits: Patient registries, clinical trial data, claims data, electronic health records (EHRs), health surveys, System for picture archiving and transmission - data from medical equipment such as CT and MRI. The user may input data via the built-in healthcare application, desktop, or online application. The factors taken into consideration for this research are the file type and size in bytes.

The encryption technique was used and stored in a multi-cloud environment based on the size of the file. To enhance security, the data are divided and kept on two different cloud storage services to prevent unauthorized access. Only one encryption and one storage were used in earlier research, which made them less safe against more sophisticated threat access assaults. Additionally, the performance of the encryption mechanism was subpar. Significant points about algorithm speed, data overhead, data integrity, and key security were included in the current study, which improved the privacy aspect and offered additional insights using this methodology. Encryption performance was insufficient beyond a certain file size while testing multiple encryption for several file types.

Therefore, this challenge was solved by choosing the appropriate file size and type and encryption procedure.

3.2 Need of Multi-Cloud

Multi-cloud approach in the study is that setups can help with growth by letting you split up jobs between different cloud systems based on their strengths. By using the unique features of each cloud service, it can improve the speed of your healthcare application. Setting up fine-grained Roy et al. (2019) access rules in multi-cloud settings. Users have to provide the correct file identity along with a symmetric key, who they are before they can access certain files. This makes sure that only authorized users can download and decrypt the files. Multi-factor authentication is used with multi-cloud setups, which adds an extra layer of protection to the decryption process. .

4 Design Specification

A detailed understanding of sensitive attribute identification and appropriate encryption algorithms is required while designing a security model tool. The high-level strategy for storing medical data in healthcare apps using a security layer is presented in detail. This section explains the performance indicators once the findings have been received.

4.1 Diagram

An overview of the application process is shown in the Figure 2

The client environment consists of Healthcare applications which utilise the main Hybrid Security layer for their application which can be used to keep data secure. From this model, the encryption and decryption process involves multi-cloud environment storage with add-on security constraints.

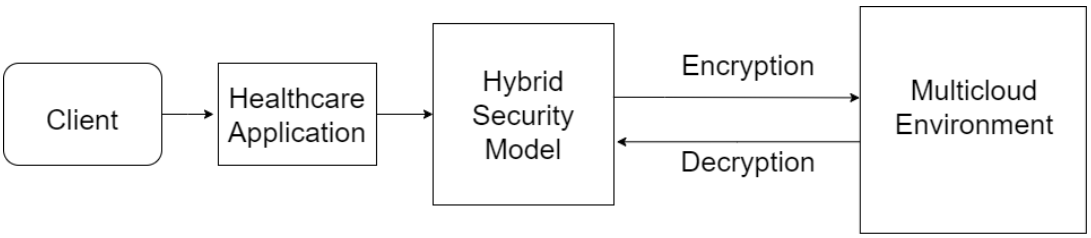

Figure 2: High Level Architecture

4.1.1 Meta Data Storage

Two kinds of cryptographic keys are used by the encrypted process to keep your data safe: one to encrypt your data and the other to maintain track of multiple files if the user has uploaded. In this study separate db file system is located on a private cloud which will consist of user ID, file ID [foreign key and primary key mapping], symmetric key, and file storage location.

4.2 Task involved in model

1. **File Type Detection:** Before applying encryption, the system will identify the type of files being processed (PDF, multimedia, text). This can be done through file extensions or content analysis.

2. **Encryption Module:** The encryption module is responsible for applying specific encryption algorithms based on the file type.

   (a) For PDF files: Use Camellia encryption algorithm.

   (b) For multimedia files: Use Triple DES encryption algorithm.

   (c) For text files: Use the ChaCha20 encryption algorithm.

3. **Multi-Cloud Integration:** The files are kept in a multi-cloud environment after they have been encrypted. The cloud storage providers used are AWS S3 and Azure Blob Service.

   (a) Text and PDF files encrypted are uploaded to a public Amazon S3 bucket.

   (b) Multimedia files encrypted are kept in an Azure Blob container on a private cloud.

4. **Key Management:** Encryption and decoding must be done securely, which requires proper key management. Keep an accessible yet secure key repository. This symmetric key is kept in the central repository for every file upload, and the patient ID will be the primary key for multiple file identification.

5. **Access Control Mechanism:** Place security mechanisms in place to guarantee that files are saved and that only authorized workers may access and edit them.

6. **Decryption Module:** When retrieving files, a decryption module is used to decrypt the files based on their type.

(a) Retrieve the encrypted PDF file from AWS S3 and decrypt it using the Camellia algorithm.

7. **Healthcare Application Integration:** Being loosely coupled structure this application service will be easy to Integrate with the healthcare application and store the files in multi-cloud as well.

8. **Logging and Auditing:** For compliance and forensic analysis, use logging and auditing capabilities to monitor changes, access, and other security-related events.

4.2.1 FlowChart Diagram

The Hybrid Security model shown in Figure 3 consists of the main part of the overall model which has the encryption mechanism along with the security posture in place. When the encryption and decryption procedure begins, the user must first choose the input data and key. Utilizing a private cloud service, the user data and key will be preserved as*metadata* and a SQL database will be created. The second phase is the characterisation of the Input data with size and type. For example consider a minimum size of 10kb TXT, 10mb PDF and 2.5gb VIDEO (mp4) will be segregated differently for next phase encryption. Only Multimedia (video) content, will be divided into small chunks to get better results. This has a primary key association with each file type which will be uniquely matched while decryption to get the proper assigned original file.

The next phase is considered as the heart of the model. Which has actual encryption namely with ChaCha20, Camellia, and TripleDES. The Camellia is mainly considered for the PDF which gives the best results when compared to other encryption. Text files are suitable with ChaCha20 encryption. The Multimedia chunks give the best performance and result with TripleDES. The next phase in this model utilises the Multi-cloud platform for better storage management and security purposes. Mainly this model consists of AWS S3 cloud service and Azure cloud. Using multiple encryption methods as per file size is a hybrid model that gives the best-performing results compared to others.

The algorithm and encryption method will be chosen based on the file size entered by the user, and the cipher text file will thereafter be saved on a cloud storage provider. If the file size is larger than 1 GB, it will be moved to Azure Cloud storage service which is a private cloud. For the multimedia, large the file the encryption process will be taken place in small partitions of 1024 byte size.

Whenever the user wants to retrieve the data after a successful upload, the decryption mechanism is necessary. Figure 4 illustrates how the decryption model works. The decryption part of the model comes into the picture with the functionality that if the user wants to download a particular file, a list of all files of that particular patient will be displayed, as per selection the file will be identified and the decryption algorithm will process on the file.

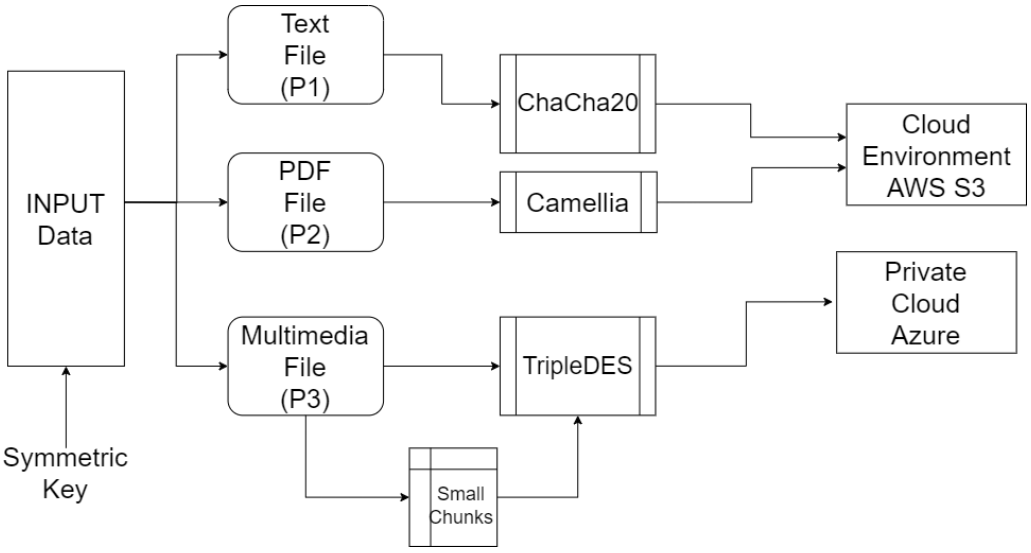

Figure 3: Hybrid Model: Encryption process of proposed system

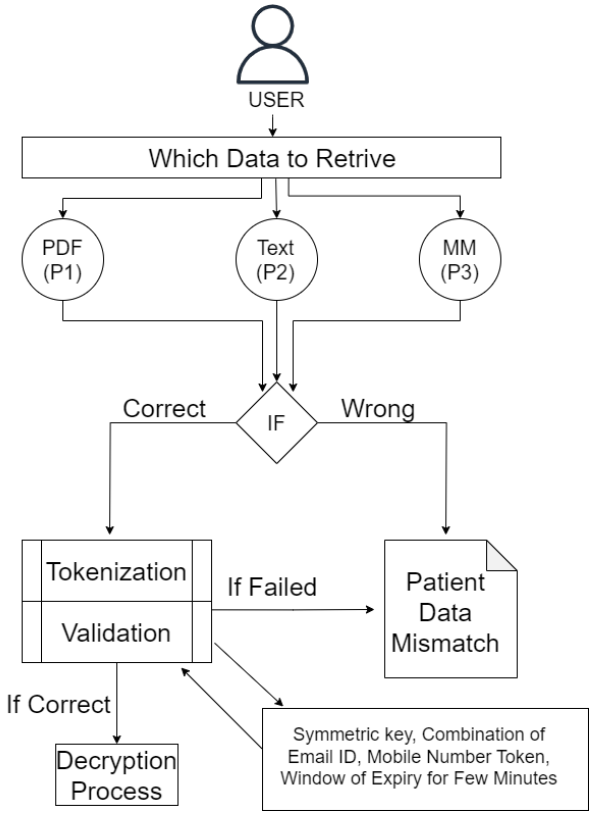

Figure 4: Decryption process of the proposed system on Multi-cloud

The file details will be cross-checked from the Metadata DB which is stored in the private Azure cloud. The IF condition, which has two sides that are CORRECT and WRONG, is inserted next to the flow. As a security measure, the tokenization and validation process, which verifies the user's identity, will come next if the type selection is accurate. This feature includes a window of expiration for a few minutes to prevent middleman attacks, a symmetric key, and a combination of email address and mobile number tokens. Should this be accurate, the decryption procedure begins and produces the appropriate input files. If the second step is unsuccessful, a notification titled "Patient Data Mismatch" will be shown.

Whenever the user wants to fetch the file, firstly user will have to specify the primary key of the patient and will have mapping with the other files of the patient's data. If the primary is assigned to every file if the user has multiple INPUT files as key matches the symmetric key needs to be provided for the decryption process. If both the authentication shows a positive match then the user will get the original file in decrypted form.

This study is based on the hypothesis that a multi-cloud system will be set up, using Amazon S3 structures and Azure Blob storage to store and handle protected files effectively. Here a multi-cloud method using both Amazon S3 and Azure Blob storage was chosen because of the need for a reliable and independent file access solution that works across multiple cloud providers. Also, it is known that Amazon S3 and Azure Blob storage are reliable and scalable when it comes to handling private data, which keeps personal files safe. since they also have their own encryption feature in-framed, dual encryption will be in the process.

4.3 Performance Indicators

The following Key Performance Indicators (KPIs) are being examined for this project:

Algorithm speed: The total speed of data exchanges within a healthcare application can be affected by the latency introduced by encryption and decryption operations. Choosing encryption techniques that are both fast and secure. There is a trade-off between speed and security that needs to be considered, as well as the unique needs of the healthcare application.

Data overhead: The size of the file after decryption shouldn't become larger than the original file. Memory requirements and processing time are the next factors to consider. Based on an evaluation of the raw file to determine its memory use and execution time, as well as the use of previously decrypted medical data to acquire the same comparison-based metrics, the comparison is made.

Data integrity: Encryption is meant to keep data secure, not compromise it. No changes can be made to the encrypted data while it is in transit or storage. Data integrity should be checked by the encryption scheme you choose or by a different mechanism like HMAC (Hash-based Message Authentication Code). This is of the utmost importance in healthcare applications since the repercussions of data tampering are severe.

Key security: Cryptography key protection is crucial to the encryption security of a

multi-cloud setup. Key-based encryption encrypts and decrypts plaintext. Your code uses a 128-bit key. Longer keys are safer but more computationally costly. In this TwoFish Algorithm research, 16-byte keys are investigated. Both Camellia and ChaCha20 have 16 bytes. Also, Key management plays a role in maintaining the metadata of the security model. For this study file system db will maintained in one public cloud service to keep track of the files of a single user. Below considered points in Table: 1

Cryptography	**Encryption algorithms**	**Multi-cloud**
Algorithm Speed	?	?
Data Overhead	?	?
Data Integrity	?	?
Key Security	?	?

Table 1: Values of Performance with a combination of encryption algorithms and multi-cloud environment

When the performance measurements are evaluated, one striking finding is found. For heavy files above 1 GB, if the encryption is processed in small chunks the computation speed is faster than bigger chunks. The encryption key bytes are then used to get reliable cipher text requirements and the processing time needed to process the results. Based on an evaluation of the raw file to determine its memory use and execution time, as well as the use of previously decrypted medical data to acquire the same comparison-based metrics, the comparison is made. we have done a comparison using an increasing range of bytes, shown graphically in Figure 5

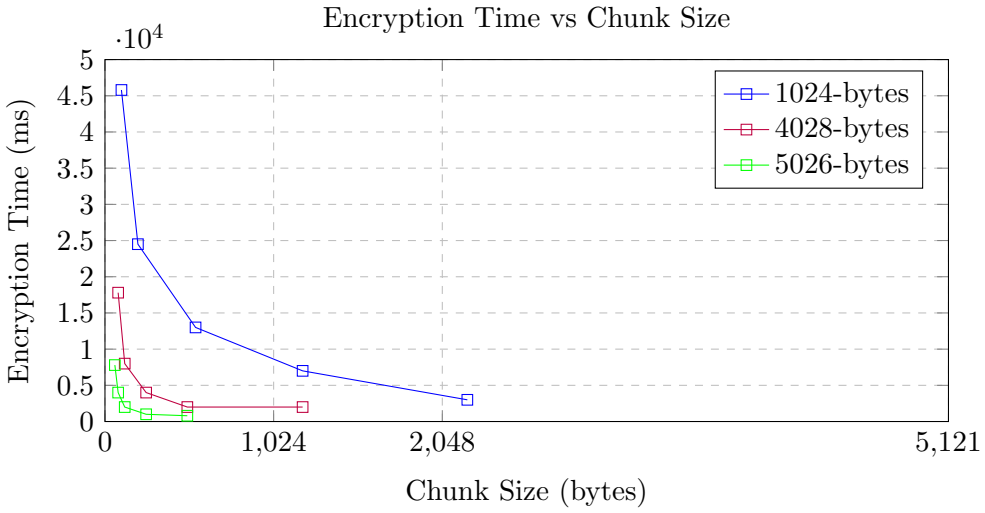

Figure 5: Reduced Size of the Graph

5 Implementation

This section explains the methods used to assess model performance as well as the proposed design's implementation plan. This section discusses and describes the application's high-level architecture and data flow. The source code for the program is shown below.

This project involves utilizing the Java programming language to create a simulator where all algorithm implementations for text, picture, audio, and video files are done. With this simulator assess the program's execution (process) time for different input sizes and then evaluate the algorithms' performance measured.

5.1 Dataset

This project utilized public datasets Pogorelov et al. (2017) ,Prabhu (2021) from the Kaggle website to perform the assessment. This research informs us about the field names of various datasets after encryption. The findings include encryption speed based on key bits and rounds 4 for all techniques using the same block size 128-bit size length.

This solution takes into consideration text, images, images, and video as its four forms of multimedia material. Consideration is given to 20 samples of file size in the simulation. Text files may range in size from 1 KB up to 10 MB. 10 image samples ranging in size from 100 KB to 1 GB and 10 video samples ranging in size from 100 MB to 3 GB are included. Every method in the experiment made use of a block size of 128 bits.

The files have been encrypted and decrypted using the Bouncy Castle Jar. Bouncy Castle makes it possible to swiftly encrypt and decode data, regardless of the content. BouncyCastle JCE[Java Cryptographic Extension] provider, which is also a free source. For this simulation, the spring boot application was coded. The Bouncy Castle Crypto package is a way to use secure methods in Java. The package is set up so that it has a lightweight API that can be used in any setting (including the brand-new J2ME) and the extra infrastructure to make the algorithms work with the JCE framework de Oliveira et al. (2018).

The experiment was run on a Mac OS M1 with 256GB SSD, 8GB RAM, 8-core CPU, and 8-core GPU. Intellija was used to create the bouncy castle jar using Java program version 8 API. Through Postman, the result is shown in JSON format.

5.2 Model Architecture

Figure 6 displays a high-level examination of a data file using a security API service. Businesses may comply with stringent file privacy regulations that mandate the preservation of patient medical history documents—which might be reports, images, or videos—by including a crypto layer into their healthcare application. Figure 6 illustrates the file processing procedure. The same encryption technique and using a single or hybrid cloud is used in a typical situation for all data kinds of file formats, which increases calculation speed and quality after decryption for larger files. It also influences network latency

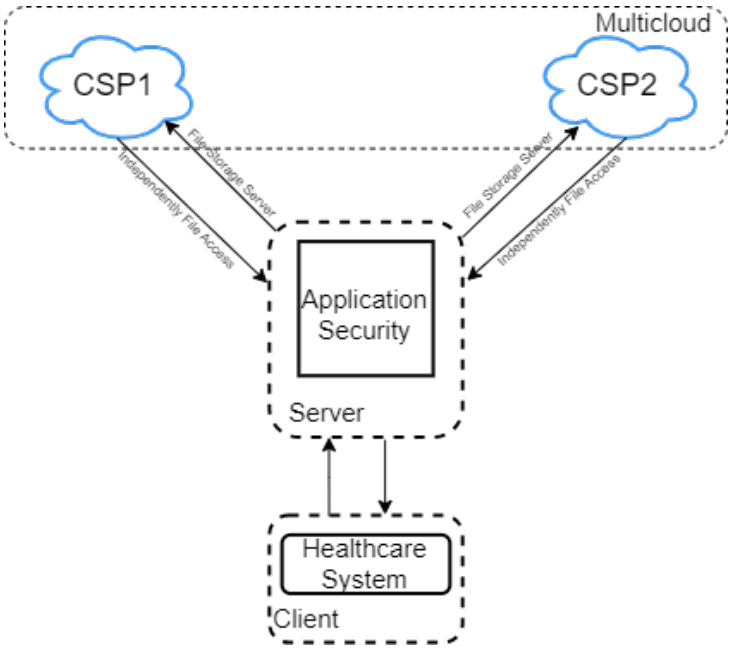

Figure 6: High Level Architecture

when files are pushed to and retrieved from cloud storage. The results of the investigation, however, indicate that no file format performs identically when carrying out certain calculations. This results in problems with the performance of the program. To solve this problem, this research developed a security API that can be added to any application.

5.3 Encryption Algorithms Main Component

The Initialization Vector (IV) is a randomly generated or pseudo-randomly generated variable that is used to initialize the encryption method before the encryption of the data. The inclusion of this element serves as a supplementary component in the encryption procedure. This IV is the main component in all Algorithms used in this study. The length of the IV is determined by the encryption technique utilized. The usual block size for Triple DES is 16 bytes (128 bits). The same block size is used in ChaCha20 and Camellia algorithms.

The encryption algorithm uses the key to encrypt and decrypt plaintext. The key in your code seems to be 128 bits (16 bytes). Longer keys are more secure but more computationally expensive. In this study for the TripleDES Algorithm, a 16-byte key length is considered. Camellia has 16 bytes and ChaCha20 has the same well.

5.4 Performance Calculations

The final three algorithms shown are encryption speeds calculated in Table 2 using the corresponding data set to get the final results. The parameters listed below are those that this research took into consideration while evaluating encryption techniques. It is computed using the file size and the key bytes that are used in each method. Small chunk sizes (bytes) for encryption are compared with the encryption algorithm's calculation time for large files as well. .

Time (Milli seconds)	ChaCha20	Camellia	Triple DES
Text dataset[1000kb]	109.7	288.3	292.3
Pdf dataset[2200kb]	49.7	48.7	78.4
Images dataset [1 gb]	N/A	6795.7	4241

Table 2: Hybrid Model with Results of Medical Dataset

5.5 Results obtained

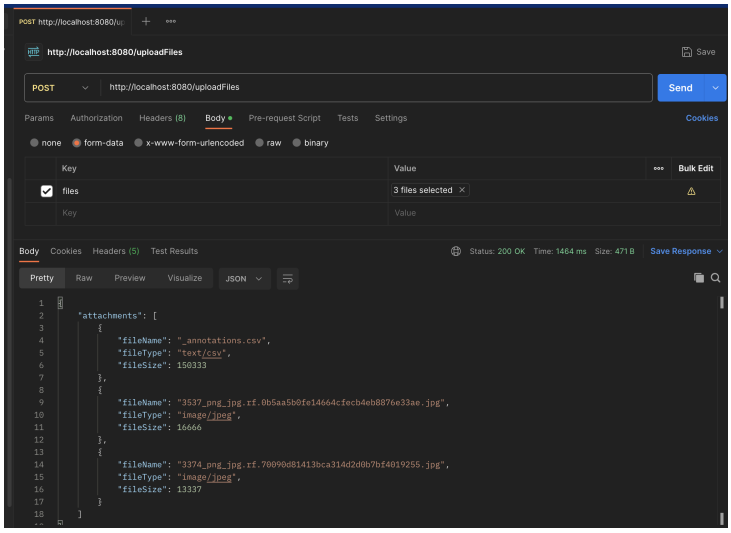

Figure 7: Postman Tool to upload file for processing

The above snapshot Figure 7 is the result received once the file is loaded and the encryption process along with file storage in multi-cloud is done in the back end.

The structured design says that the protected files will be kept in a multi-cloud environment. Once the files have been encrypted, they will be sent to cloud services, as shown in Figure 8. To do this experiment, an AWS S3 bucket is picked. For identification, the "Encrypt" term is included before the original name.

6 Evaluation

6.1 Comparison with Data encryption approach

In earlier research, it was thought of as a structure where file data would be divided using two fish cryptography and an encryption technique before being transferred to a multi-cloud environment. We overcame this issue by switching the approach of encrypting the file instead of data inside the file. During implementation, fetching the parts from a multi-cloud environment was laborious and it was observed that the data inside the file was not fully decrypted and that some parts were scrambled. As a result, the experiment failed.

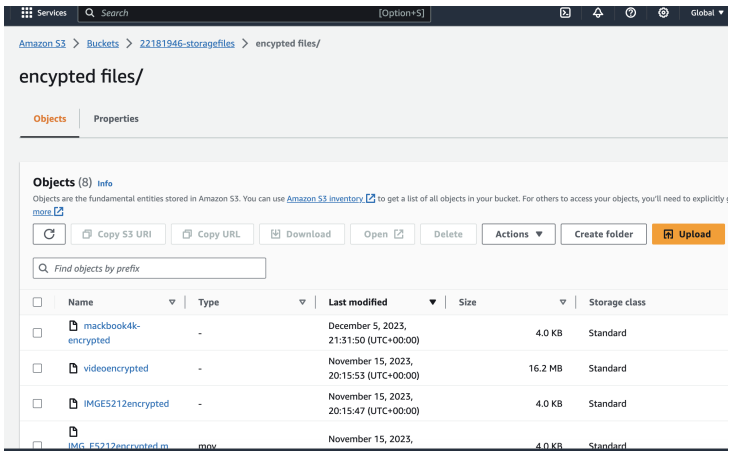

Figure 8: Files storage in AWS S3 bucket

6.2 Comparison with model of cloud resource

The addition of comparative studies of this model with the usage of encryption with a single cloud, hybrid cloud, or other multi-cloud suppliers is going to be made. It was decided in the second trial to use the same technique for all file kinds, including text, pdf, and multimedia files. However, when the file size was more than 1000 kilobytes, the performance of the encryption process became poor, resulting in a worse quality of the decrypted file in comparison to the original. It also resulted in the file size increasing to a ratio of twice.

6.3 Comparison with Encryption algorithms with various file types

Camellia and ChaCha20 demonstrate competitive speeds, particularly for smaller file sizes. Figure 9 These algorithms could be well-suited for applications with stringent speed requirements.TwoFish consistently outperforms other algorithms, boasting minimal encryption times, which can be attributed to its efficient design and implementation. Triple DES, while offering decent performance, exhibits higher encryption times compared to the other algorithms. It may be a suitable choice depending on the specific security requirements of the application. Below shown in Table 3 are a range of experiments with various algorithms.

Encryption time for TEXT files							
Time (Milli seconds)	1 KB	10kb	100kb	200kb	400kb	600kb	1000kb
TwoFish	0	56	115.29	220.6	467.8	490.6	512.6
Camilliea	0	76.8	225.98	260.2	271.7	278.3	288.3
ChaCha20	0	34.12	90.67	93.6	99.1	102.3	109.7
Triple DES	1	78.1	189	206.7	223.1	246.8	292.3

Table 3: Average Encryption time for TEXT files

Performance Across File Sizes, Camellia's capacity to adapt to different data quantities is shown by its steady performance even as PDF file sizes rise.Figure 10 Camellia's

Figure 9: Encryption Time vs File Size Graph for TEXT files

encryption duration stays within a reasonable range, making it appropriate for use in applications involving varying sizes of PDF files. Below shown in Table 4 are a range of experiments with various algorithms.

Encryption time for PDF files							
Time (Milli seconds)	10 KB	100Kb	200Kb	500Kb	900Kb	1500Kb	2200kb
TwoFish	1.1	35	35.5	36.3	41.5	48.7	55.2
Camilliea	1	31	31	33.2	34.4	37.8	48.7
ChaCha20	1	33	33.1	36.2	40.6	44	49.7
Triple DES	1	41	46	52	58.2	67.2	78.4

Table 4: Average Encryption time for PDF files

Across all file sizes, Triple DES reliably produces excellent encryption speeds, ranging from 2.5 milliseconds for 10MB to 5781 milliseconds for 3GB. It is a trustworthy option for applications with a range of data sizes because of its performance, which is noticeably competitive even for bigger multimedia files. Among the encryption algorithms, Triple DES Figure 11 is particularly strong, exhibiting reasonable encryption times for a variety of multimedia file sizes. It provides programs working with multimedia datasets of different sizes with flexibility and dependability. Larger datasets may provide difficulties for the Camellia and Serpent algorithms, despite their effective encryption for smaller multimedia files. Below shown in Table 5 are a range of experiments with various algorithms.

Encryption time for MULTIMEDIA files							
Time (Milliseconds)	10mb	200Mb	500Mb	800Mb	1Gb	2.2Gb	3Gb
Camilliea	1.3	2.5	2818	6795.228086	7109	7734	8101
Serpent	3.8	4.7	2589	5152.690123	5789	6487	6578
Triple DES	2.5	3.2	1930	4181.11	4245	5678	5781

Table 5: Average Encryption time for MULTIMEDIA files

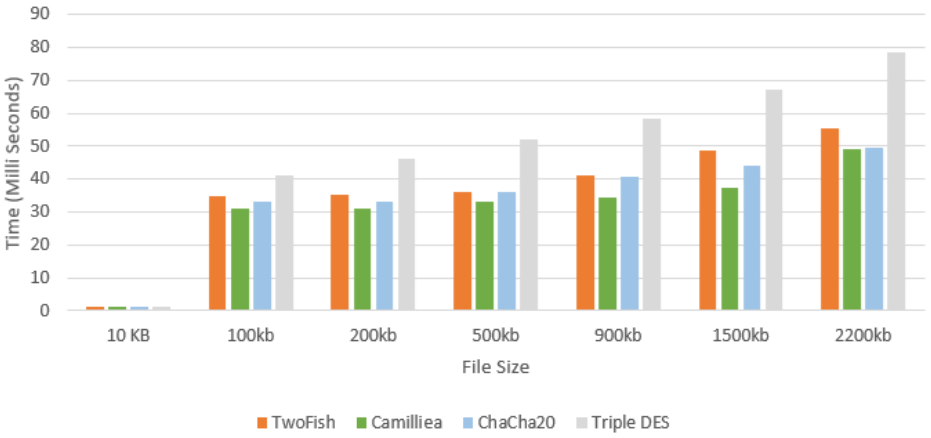

Figure 10: Encryption Time vs File Size Graph for PDF files

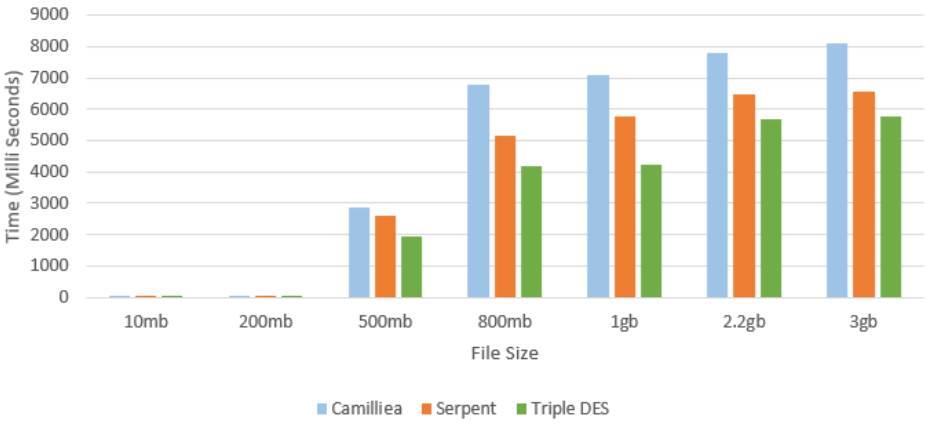

Figure 11: Encryption Time vs File Size Graph for MULTIMEDIA files

6.4 Comparison with Medical Dataset with different encryption models

Using the KSM blood bank dataset, this research compared our results to those of Maathavan and Venkatraman (2022). Since our analysis also incorporated data overhead and data integrity elements, the performance rate in their model—which uses the AES-GCM technique for encryption—is 12.43 greater than our study's, which is 10.97.

The research conducted by Pushpa (2020) compares performance measures to verify data quality both before and after encryption using photos from a medical dataset. Their hybrid encryption strategy encrypts data using the blowfish and two fish algorithms, yielding an MSR value of 0.10 which indicates image quality. In our comparison analysis, we were able to get a value of 0.57.

In this paper, Naeem et al. (2022) uses a synthetic dataset, the PHI medical dataset, which contains 5000 patient reports in PDF format. For this study, we selected files ranging in size from 10 KB to 200 KB. When the computation's performance factor is

compared, the Camellia algorithm performs better than multi-encryption models; additionally, their study does not take data overhead or file size increment factor into account. Below shown in Table 6 are parameters that show the performance indicators achieved by the existing model

Schemes	**Algorithm Speed**	**Data Overhead**	**Data Integrity**	**Key Security**
Paper1	×	✓	×	✓
Paper2	✓	×	✓	×
Paper3	×	×	×	✓
Proposed Model	✓	✓	✓	✓

Table 6: Comparison of papers with compare to performance indicator

6.5 Discussion

It is shown via cryptography and multi-cloud analysis that the study's key security, data overhead, and encryption speed histogram have a uniform distribution. 7.99951 is the hybrid model's entropy value, which is near the optimum value. The proposed cryptographic framework has a large secret keyspace and an effective encryption effect. Additional research demonstrates that the suggested paradigm may enhance the security layer of healthcare applications from a variety of angles. The suggested scheme's execution time is tested using a few files of varying sizes. We observe that compared to typical security models, our technique needs a much shorter computation time Figure12.

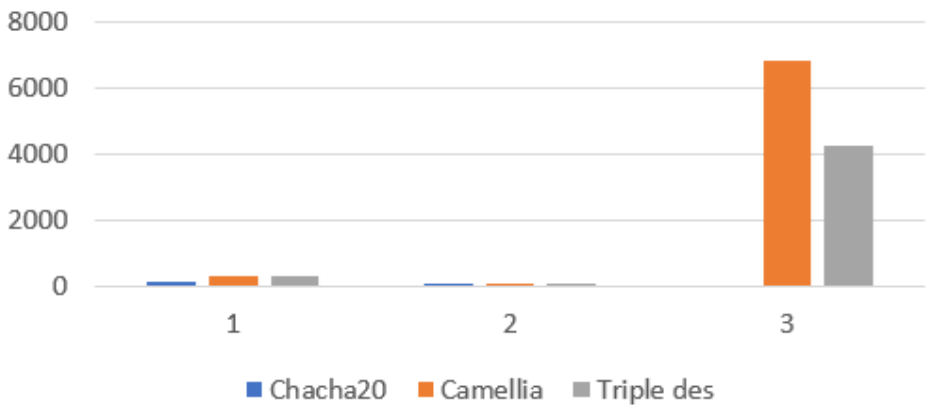

Figure 12: Average encryption running time for all ranges file size

7 Conclusion and Future Work

To safeguard diagnostic data in healthcare applications, this research introduced a novel hybridization of the data encryption paradigm. That combines the best features of Chacha20, Camellia and the TripleDES encryption algorithm—into a single security layer procedure. The given methodology encrypts sensitive information first and then stores it in a Multicloud environment. The suggested model's performance has been evaluated using several performance metrics, and it has been compared against various benchmark file types. Using user identity and multi-cloud storage, which eliminates area restrictions and provides vast storage capacity, the suggested approach remarkably achieves great results with minimal network latency.

This study can also have an improvement area in the key management area where the metadata is stored. Examine the suggested model's performance and scalability in big healthcare systems with a lot of diagnostic data. Think about possible parallelization or optimization techniques to manage growing data quantities with minimal delay. Work together with healthcare organizations and institutions to put the suggested model into practice and assess it in practical situations. Pilot projects or collaborations might be used to get input and evaluate the model's applicability and efficacy in a clinical setting.

References

Alam, A. B. M. B., Fadlullah, Z. M. and Choudhury, S. (2021). A resource allocation model based on trust evaluation in multi-cloud environments, *IEEE Access* **9**: 105577–105587.

Anwarbasha, H., Sasi Kumar, S. and Dhanasekaran, D. (2021). An efficient and secure protocol for checking remote data integrity in multi-cloud environment, *Scientific Reports* **11**(1): 13755.
URL: *https://doi.org/10.1038/s41598-021-93073-3*

Aski, V. J., Gupta, S. and Sarkar, B. (2019). An authentication-centric multi-layered security model for data security in iot-enabled biomedical applications, *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*, pp. 957–960.

de Oliveira, C., Turnquist, G. and Antonov, A. (2018). *Developing Java Applications with Spring and Spring Boot*, Packt Publishing.
URL: *https://books.google.ie/books?id=EuLIvAEACAAJ*

Devi, R. and Chamundeeswari, V. (2020). Triple des: Privacy preserving in big data healthcare, *International Journal of Parallel Programming* **48**.

Dr Ramalingam Sugumar, K. R. (2018). *EDSMCCE : Enhanced Data Security Methodology for Cloud Computing Environment* **2456-3307**: 40–46.

Elhoseny, M., Ramírez-González, G., Abu-Elnasr, O. M., Shawkat, S. A., Arunkumar, N. and Farouk, A. (2018). Secure medical data transmission model for iot-based healthcare systems, *IEEE Access* **6**: 20596–20608.

Lee, B.-H., Dewi, E. K. and Wajdi, M. F. (2018). Data security in cloud computing using aes under heroku cloud, *2018 27th Wireless and Optical Communication Conference (WOCC)*, pp. 1–5.

Maathavan, K. and Venkatraman, S. (2022). A secure encrypted classified electronic healthcare data for public cloud environment, *Intelligent Automation Soft Computing* **32**: 765–779.

Naeem, U. H., Bilal, M., Syed, F., Rasheed, K. and Saad, S. A. (2022). A multilayer encryption model to protect healthcare data in cloud environment, *2022 International Conference on Data Analytics for Business and Industry (ICDABI)* pp. 42–48.
URL: *https://api.semanticscholar.org/CorpusID:256877209*

Pogorelov, K., Randel, K. R., Griwodz, C., Eskeland, S. L., de Lange, T., Johansen, D., Spampinato, C., Dang-Nguyen, D.-T., Lux, M., Schmidt, P. T., Riegler, M. and Halvorsen, P. (2017). Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection, *Proceedings of the 8th ACM on Multimedia Systems Conference*, MMSys'17, ACM, New York, NY, USA, pp. 164–169.

Prabhu, S. (2021). Blood bank directory - india.
URL: *https://www.kaggle.com/datasets/sachinprabhu007/blood-bank-directory-india*

Pushpa, B. R. (2020). Hybrid data encryption algorithm for secure medical data transmission in cloud environment, *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)* pp. 329–334.
URL: *https://api.semanticscholar.org/CorpusID:216104692*

Rashidi, B. (2021). Flexible and high-throughput structures of camellia block cipher for security of the internet of things, *IET Computers Amp; Digital Techniques* **15**: 171–184.

Roy, S., Das, A. K., Chatterjee, S., Kumar, N., Chattopadhyay, S. and Rodrigues, J. J. P. C. (2019). Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications, *IEEE Transactions on Industrial Informatics* **15**(1): 457–468.

Sajjan, R. and Ghorpade, V. (2019). Gcm-aes-vr : A scheme for cloud data confidentiality and authenticity.

Weir, C., Dyson, A. and Prince, D. (2023). A lot less likely than i thought: Introducing evidence-based security risk assessment for healthcare software, *2023 IEEE Secure Development Conference (SecDev)*, pp. 156–170.