National College of Ireland

# Design and Evaluation of a Cloud-Based Secure Medical Image Storage and Sharing Framework using Cloud Security Principles

MSc Research Project
MSc in Cloud Computing


KALAL SRAVAN KUMAR GOUD
Student ID: X23342943

School of Computing
National College of Ireland

Supervisor: Shivani Jaswal

**National College of Ireland**
**Project Submission Sheet**
**School of Computing**

National College of Ireland

| | |
|---|---|
| **Student Name:** | KALAL SRAVAN KUMAR GOUD |
| **Student ID:** | X23342943 |
| **Programme:** | MSc in Cloud Computing |
| **Year:** | 2025 |
| **Module:** | MSc Research Project |
| **Supervisor:** | Shivani Jaswal |
| **Submission Due Date:** | 11/12/2025 |
| **Project Title:** | Design and Evaluation of a Cloud-Based Secure Medical Image Storage and Sharing Framework using Cloud Security Principles |
| **Word Count:** | 8858 |
| **Page Count:** | 19 |

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:** | KALAL SRAVAN KUMAR GOUD |
| **Date:** | 11th December 2025 |

## 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): | |

# Design and Evaluation of a Cloud-Based Secure Medical Image Storage and Sharing Framework using Cloud Security Principles

KALAL SRAVAN KUMAR GOUD
X23342943

**Abstract**

Healthcare has transitioned to digital form which has created the congruence of many challenges to how sensitive medical image data will be stored, maintained, and secured. Security is of utmost importance when considering that data breaches occur unexpectedly and you have to comply with many more laws and regulations than before. Additionally, the sheer volume of medical imaging data that must be stored and made available increases the need for security solutions that scale to provide adequate protection for all medical images. This particular project describes the design, build, test and evaluation of a secure medical image sharing and storage system in the cloud, using Amazon Web Services (AWS).

The secure medical image sharing and storage system is based upon a multi-layered security architecture that uses the core cloud principles of security such as client-side envelope encryption using the AWS Key Management Service (KMS), Amazon Simple Storage Service (S3) for durable storage of medical images, Amazon DynamoDB for structured metadata storage, and the use of Amazon CloudWatch to provide comprehensive auditing of all medical images processed. The system has been thoroughly tested and evaluated by both functional validation and real-world data processing of DICOM and TIFF formats, and by conducting quantitative performance analysis of the developed framework.

The framework is confirmed to function correctly and securely manages sensitive medical image data. The benchmark for performance indicates that the average throughput for the combined process of encrypting and uploading is 0.42 MB/s, while the average throughput for downloading and decrypting is 0.48 MB/s, which demonstrates that the proposed solution has sufficient performance characteristics for clinical use. This research has developed a practical framework and documentation for building a secure and efficient cloud-based medical image management system.

## 1 Introduction

With the rise of technological digital imaging, an increasing number of sensitive patient data are being created and stored in different formats by digital imaging technologies/techniques such as computed tomography (CT), magnetic resonance imaging (MRI), x-ray, and others. This has led to a growing number of patient data that are required for the diagnosis and treatment of patients; however, it has also resulted in many challenges

in collecting, managing and securing this data. In most cases, when healthcare providers utilize traditional on-premises storage methods to store this data, those storage methods tend to be expensive to utilize, not easily scalable, and present barriers for healthcare provider collaboration (Shakor and Khaleel (2024)). Many traditional on-premises storage methods are also unable to accommodate the volume of data being generated, which creates data silos and prevents healthcare providers from being able to exchange information in an efficient manner when providing essential patient care and conducting advanced medical research (Vashishth et al. (2025)). Sensitive Protected Health Information (PHI) and other electronic data are frequently stored in on-premises records; however, when migrating this data to a cloud infrastructure, it poses a considerable risk for security and privacy breaches and will require comprehensive risk mitigation prior to complying with HIPAA and GDPR regulations (Theriault-Lauzier et al. (2024)). There is currently a lack of a practical, reliable, and auditable method to move PHI to the cloud and allow healthcare providers to utilize these innovative technological solutions without compromising the confidentiality and integrity of patient information (Yan et al. (2024)). This paper addresses the design and evaluation of a secure, auditable, and high-performance cloud-based architecture for storing and distributing medical images while adhering to cloud security principles. Security controls must be implemented in a way that does not hinder clinical workflow processes.

## 1.1 Research Question

How can a cloud-based framework be designed and implemented to securely store, manage, and share medical images while ensuring confidentiality, integrity, availability, and auditability through cloud security principles?

## 1.2 Research Objective

The following objectives supported the overall aim of the project:

1. Develop a cloud-native infrastructure following principles such as Defence in Depth, Least Privilege, and end-to-end encryption of data throughout its lifecycle.

2. Build and evaluate a prototype implementation using widely available cloud services and programming languages to provide a concrete realisation of the designed architecture.

3. Evaluate the security characteristics of the Cloud-Native Architecture (CNA) to verify that data is protected using strong encryption and auditing mechanisms to ensure confidentiality, integrity, and non-repudiation.

4. Evaluate the performance of core architectural functions (image upload, download, encryption, and decryption) to determine feasibility in real-world clinical environments.

## 1.3 Research Contribution

This research provides an end-to-end implementation and evaluation of a secure medical imaging management system. Unlike theoretical or high-level proposals, this work delivers a fully implemented architecture with performance benchmarks derived from empirical

experimentation. As illustrated in the conceptual architecture diagram. The final system integrates multiple managed cloud services to establish a comprehensive security posture while reducing operational overhead for healthcare organizations and enabling a focus on patient care.

## 1.4 Report Structure

This report is structured to present the design, implementation, and evaluation of a secure cloud-based medical image management framework. Chapter 1 Introduces the research background, research question, objectives, and contributions. Chapter 2 Reviews related work on cloud security, cryptography, and medical data protection. Chapter 3 Describes the research methodology adopted. Chapters 4 and 5 Present the system design and implementation. Chapter 6 Evaluates the framework through functional, compatibility, and performance testing, followed by discussion. Chapter 7 concludes the report and outlines future work.

# 2 Related Work

Numerous studies on secure management of electronic medical records in the cloud have been undertaken, including studies related to different types of encryptions, authentication methods or structures, new technologies (e.g., Blockchain & Artificial Intelligence) to secure and manage healthcare data and other related technologies. This part presents a critical review of scientific studies related to this area, which allows us to place this project within the current literature and establish an understanding of current approaches, as well as highlight an area of research that this study will address.

## 2.1 Foundational Cloud Security Strategies for Healthcare Data

A robust security posture for cloud-based healthcare systems cannot be achieved through a single mechanism but requires a comprehensive, multi-layered strategy. Vashishth et al. (2025) provide a foundational overview of such strategies, emphasizing the need to protect sensitive patient data at all stages of its lifecycle. Their work outlines a defense-in-depth approach that integrates encryption for data at-rest and in-transit, stringent identity and access management, and continuous network monitoring to detect and mitigate threats.

This strategic perspective underscores that any secure framework must be built upon a combination of preventative, detective, and corrective controls. The principles they discuss—such as data minimization and the principle of least privilege—are critical for establishing a baseline of security upon which more advanced and specific mechanisms can be built. This broad framework serves as a vital starting point for understanding the essential components required to protect sensitive data within a cloud ecosystem.

### 2.1.1 Granular Access Control for Medical Big Data

While foundational strategies are essential, the unique requirements of healthcare demand more sophisticated access control mechanisms than standard models provide. In a clinical environment, access rights are dynamic and context-dependent, requiring a system that can enforce complex policies. Addressing this, Alabdulatif et al. (2023) propose a

novel cloud-enabled access control model designed specifically for the security and privacy of medical big data. Their model moves beyond traditional role-based access control (RBAC) to a more granular, attribute-based approach. This allows for the creation of dynamic access policies that consider multiple factors, such as the user's role, the data's sensitivity level, the purpose of access, and even temporal constraints. By implementing such a fine-grained model, healthcare organizations can ensure that data is only accessible by authorized personnel on a strict need-to-know basis, significantly reducing the risk of unauthorized data exposure and aligning with privacy regulations like HIPAA.

### 2.1.2 Privacy-Preserving Computation with Homomorphic Encryption

A significant vulnerability in cloud environments arises when data must be decrypted for processing, such as during analysis by machine learning algorithms. This moment of decryption creates a window where sensitive information is exposed. To mitigate this risk, advanced cryptographic techniques are being explored. Kartit (2022) presents a novel approach utilizing homomorphic encryption to secure medical images within cloud computing environments. Homomorphic encryption is a transformative technology that allows for computations to be performed directly on encrypted data (ciphertext) without needing to decrypt it first. The result of the computation remains encrypted and can only be accessed by an authorized party with the decryption key.

The application of this technique, as explored by Kartit (2022), enables the secure delegation of computationally intensive tasks, like image analysis, to untrusted cloud servers. This ensures that the content of the medical images remains confidential throughout the entire processing pipeline, offering a powerful method for preserving patient privacy during advanced data analysis.

## 2.2 Cryptographic Solutions

Numerous researchers have focused their attention on developing cryptographic solutions for the purpose of safeguarding data confidentiality. Kartit (2022) proposed a new technique called homomorphic encryption (HOE) which uses only the encrypted version of medical images and enables calculation on data without decrypting them. Although HOE has great potential for performing secure analyses without compromising patient privacy, it often suffers from an excessive overhead associated with performing the calculations on the encrypted files before they can be analyzed; thus making them impractical for use in a clinical environment where large sets of images are stored and need to be retrieved in a timely fashion. On the other hand, Lin and Others (2025) have researched various chaotic methods for encrypting medical images as well. While these techniques provide good theoretical security, they can be confusing because they don't follow the same standards as traditional encryption methods and because of how difficult it is to manage the keys that would be needed for the encryption. This project uses a standard industry method known as AES-256 with an envelope encryption model utilizing the AWS KMS for key management to create a framework that offers a reasonable balance between a high level of secure protection and a practical, scalable system that can be managed easily.

## 2.3 Access Control

Another major area of research related to securing cloud-based health data is Access Control. Alabdulatif et al. (2023) present a new method of Cloud-enabled Access Control

based on Policy-enforced Granularity for protecting the privacy and security of medical Big Data. The authors concentrate on developing policy-based, granular rules for establishing access control to health-related data in complex healthcare scenarios. The authors state, however, that their proposed Access Control Model is purely theoretical at this time and lacks a complete implementation and evaluation to demonstrate how it will actually perform in a real-world environment.

In addition, Punia and Others (2024)report on a systematic review of existing Blockchain-based Access Control Models, indicating their potential as a means of establishing decentralized and immutable "Audit Trails" for access logs of these systems. These models offer good non-repudiation capabilities, but the integration of blockchain into a system generally results in increased complexity, cost, and latency, which may limit their usefulness in certain scenarios. A practical way to create a cloud-based health data access control framework is to implement an "RBAC" role-based access control model, using S3 Pre-signed URLs to provide secure, limited-time access without introducing the overhead typically associated with complex systems; this offers the same level of security and auditability that these systems provide, while being far easier to use and manage within many common online clinical collaboration environments.

## 2.4 Integration of Cloud Computing

The integration of cloud computing with other advanced technologies such as IoT (Internet of Things) and artificial intelligence (AI) has received significant research attention. In analyzing the principles of IoT security, Hossain and Others (2024) examined IoT security principles in-depth to highlight how medical imaging technologies are increasingly becoming devices that connect to external networks, and therefore expose themselves to security risks. Research by Lin and Others (2025) and Nazir and Kaleem (2023) utilize deep learning models to perform medical image analysis; these models require a high level of computational power typically supplied through a cloud service. As shown in this body of work, there is a strong need for robust and efficient data pipelines to facilitate the input of data into AI-based models; meeting this foundational need is a key objective of the framework that we present here. Lastly, Theriault-Lauzier et al. (2024) proposed a responsible AI framework for implementing AI solutions at points of care and stress the importance of having a secure, auditable, and reliable data infrastructure in place.

## 2.5 Cloud and Medical Data Security

Recently, several researchers have highlighted the increasing importance of developing integrated solutions to address the complexities of Cloud-based medical systems, as well as the need to address the continued presence of fragmented or siloed frameworks for Cloud security, including issues related to Authentication, Data Isolation and Regulatory Compliance Chauhan and Shiaeles (2023) and Shakor and Khaleel (2024). Ali and Others (2024) have provided a more comprehensive analysis of existing Risk Assessment methodologies for Cloud computing and argued that Healthcare Systems do not possess any authoritative guidance for evaluating Cryptographic Strength, Access Control Maturity or Data Handling Vulnerabilities. Numerous Cryptographic Models have recently been introduced, such as Cloud-Based DNA Cryptography Kairi and Bhadra (2025) Quantum-Enabled Algorithms for Privacy-Preserving Medical Image Processing Yan et al. (2024) and Secure Sharing Techniques to Distribute Encrypted Health Records on Multiple

Cloud Nodes Mahammad and Rani (2025). However, many of these solutions introduce excessive processing overhead and require advanced infrastructure and/or expertise to implement, which may limit their increasing availability and value to Cloud-based Applications. According to a report by Kondylakis and Others (2023) large-scale medical imaging initiatives within the EU are currently facing challenges when it comes to the establishment of secure and auditable Data Pipelines for both developing and deploying Artificial Intelligence (AI) models. Similarly, the articles by Shakor and Khaleel (2024) and Khaleel (2024) and Ali (2025) state that developing Scalable Cloud-based Architectures to support the use of Deep Learning (DL) is essential to enable use of DL. Lastly, the research of Montagnese et al. (2025) demonstrates that equitable and value-adding healthcare applications such as platforms to support the care of persons with dementia depend on cloud solutions that provide a trusted environment, including encryption, access control and governance. Collectively, these papers showcase the presence of technology and developments related to Data Protection, Access Control, and AI Imaging; however, there are currently no integrated frameworks to validate performance of deployment with regards to all three aspects identified above. Hence, this current Study endeavors to fill an important gap found in Literature, including developing a unified, implemented, and validated Performance Framework to enable the integration of all the Above-mentioned aspects of Security into AI Imaging Systems.

## 2.6 Comparative Analysis

In comparison to the previously published (in Table 1), which covers a wide range of contributions and weaknesses in published literature as well as how the proposed work relates to that previous literature, it is evident from all the previously published works that in general, most of the works have proposed some type of novel individual component or components (such as an encryption algorithm or a theoretical model for access control) but there is a clear lack of published works that have demonstrated a complete, developed, and tested implementation of a security-related application. Most of the works that are currently published are either only concepts or do not have sufficient performance benchmarks to provide healthcare practitioners and IT managers with the type of real-world framework they need to evaluate each of the proposed solutions. Consequently, the proposed work will fill a significant void by supplying an applicable, functional implementation and extensive evaluation of its integrated security features, as well as its overall operational performance, making It an immensely valuable and significant addition to the literature in this area.

existing research has attempted to tackle the secure medical image management issue, providing useful knowledge, though still leaving behind some of the practical gaps more noteworthy. Kartit (2022) while demonstrating the prospects afforded by homomorphic encryption for confidential computation, suffers from the disadvantages of extremely heavy computational overhead and hence is impractical for large clinical datasets. Lin and Others (2025) discuss chaotic encryption models with attractive security guarantees. However, these methods are still not standardised, hence difficult to put into practice on a large scale. Proposed access controls by Alabdulatif et al. (2023) and Punia and Others (2024) are policy-based and blockchain-driven, and both some parts remain never fully implemented in the world today, with performance issues measurable. Mahajan (2022) similarly presents a conceptual cloud–blockchain EHR architecture, yet operational validation of their proposed framework is never provided. Lin and Others

(2025) and Kondylakis and Others (2023) see the need for secure infrastructures to be able to support AI-driven imaging workflows; however, neither provides a working solution. Hossain and Others (2024) and Ali and Others (2024) provide reviews of IoT vulnerabilities and cloud threats but fall short of proposing deployable healthcare architectures. This study directly addresses these limitations through an operational cloud-native framework that embraces industry-standard AES-256 encryption, AWS KMS key management, RBAC-based access control, and CloudWatch auditing. Unlike previous theoretical models, it has been rigorously tested for scalability and optimization for real clinical imaging workloads.

Table 1: Summary of related work and how this project addresses gaps

| **Reference** | **Core Technology / Method** | **Security Focus** | **Strengths** | **Limitations / Research Gap** | **How This Project Addresses the Gap** |
|---|---|---|---|---|---|
| Kartit (2022) | Homomorphic Encryption (HE) | Confidentiality during computation | Enables encrypted computation. | Very high computational overhead; impractical for large datasets. | Uses AES-256 envelope encryption for secure, scalable processing. |
| Lin et al. (2025) | Chaotic Encryption Models | Image-level confidentiality | Novel cryptographic strength. | Not standardised; difficult key management. | Uses standard AES-256 + AWS KMS for interoperability. |
| Alabdulatif et al. (2023) | Policy-Based Access Control | Granular access permissions | Fine-grained control for clinical workflows. | Theoretical model; no implementation. | Implements AWS-based RBAC validated by workflow testing. |
| Punia et al. (2024) | Blockchain-Based Access Control | Auditability, decentralised logs | Immutable, tamper-evident audit trails. | Blockchain introduces high latency and complexity. | Uses CloudWatch for fast, scalable auditing. |
| Mahajan (2022) | Blockchain + Cloud EHR Architecture | Secure EHR sharing | High-level secure architecture. | Conceptual only; no real evaluation. | Provides a fully operational and tested architecture. |
| Li et al. (2023) | Deep Learning for Medical Image Analysis | Secure AI pipelines | Shows need for large secure datasets. | Focuses only on AI models, not infrastructure. | Provides secure cloud-native pipeline for AI workloads. |

Continued on next page

Table 1 – continued from previous page

| Reference | Core Technology / Method | Security Focus | Strengths | Limitations / Research Gap | How This Project Addresses the Gap |
|---|---|---|---|---|---|
| Hossain et al. (2024) | IoT Security Principles | Device-level security | Highlights IoT risks in imaging devices. | No cloud storage / access architecture. | Provides secure backend that integrates safely with IoT imaging systems. |
| Ali et al. (2024) | Cloud Security Risk Assessment | Cloud vulnerabilities | Comprehensive risk analysis. | No validated architecture or implementation. | Implements encryption, RBAC and auditing with AWS services. |
| Kairi & Bhadra (2025) | DNA Cryptography | Novel protection model | Innovative ultra-secure encoding. | Too computationally expensive. | Uses efficient AES-256 envelope encryption suitable for clinical workloads. |
| Kondylakis et al. (2023) | AI Data Infrastructure in EU Imaging | Governance & secure pipelines | Identifies needs for large-scale medical imaging data. | Provides no full-stack implementation. | Demonstrates functioning cloud-native imaging pipeline with performance evaluation. |

# 3 Methodology

This study utilized a methodical approach known as a constructive research methodology that helps researchers create and then evaluate new information technology (IT) artefacts for the purposes of solving real-world problems. The conceptualisation of an IT artefact is an ongoing activity and will be carried out in a systematic manner, typically in the order of creation, implementation of that artefact, and the evaluation of it against its defining criteria. A constructive research methodology is particularly appropriate in the areas of computer science and engineering research because the outcome will be a physical or tangible product (e.g. a system, framework, or model). The research structure consists of three defined phases which are sequential in that Phase 1 will be used as a foundation for Phase 2 (Implementation), and Phase 2 provides the foundation for Phase 3 (Evaluation). Each phase will be designed to ensure the resulting framework has a solid theoretical base and follows established guidelines as well as a rigorous implementation based on its requirements and objectives.

## 3.1 Architectural Design

Concept and design of the System Architecture was the objective of Phase 1 of the project. This included looking at accepted Cloud Security best practices as described in the industry and AWS white papers on security, and determining how they should be applied to create an effective Architecture. Some of the major concepts that drive the Architecture are 1) Defense-in-Depth: the use of multiple layers to provide multiple levels of Security controls; 2) AWS Shared Responsibility Model: clarity of Responsibility for Security; 3) Principle of Least Privilege: only granting Permissions to Entities based on the specific need for those Permissions to complete their assigned tasks; 4) Data Encryption: the requirement that all Data, no matter its state (in-transit or at-rest), must be encrypted. The Architecture was designed with a focus on taking advantage of the benefits that AWS Managed Services provide, where available. The design ensures that it takes advantage of the technical Security Expertise of the AWS Provider. Selecting certain services like: 1) Amazon S3 (Storage); 2) AWS KMS (Cryptographic Key Management); 3) Amazon DynamoDB (Structured Metadata Management); 4) Amazon CloudWatch (Log Management) was important to define the interaction patterns between each of these Services to create a cohesive and secure Data Lifecycle.

## 3.2 Framework Implementation

The second phase turns the architecture design into an operational prototype using Python, as it has an established ecosystem and ample supporting material for AWS cloud functionality through the Boto3 SDK, as well as many libraries for cryptography. For software engineering purposes, the implementation was structured as independent modular components to allow for each component to manage a certain portion of the solution's functionality. The end result of this engineering practice was a set of modules for the following areas: configuration, key management, encryption, S3 access, managing metadata, managing access control, and auditing. The ability to develop modules independently, perform unit tests independently for each module, and to allow for future enhancements and maintenance will provide for a more manageable solution moving forward. To support the development of all the core modules, a series of scripts were also developed for automating the deployment of the solution, demonstrating its functionality, and performing evaluation testing on the software.

## 3.3 Fully Loaded Evaluation

The final and most important phase was the assessment of the base system framework through use of the comprehensive evaluation plan in a multi-dimensional way (see below) so that all aspects of the base system were considered. For example, both functional and non-functional requirements of the base system were evaluated; thus, evaluation of the base system framework consisted of three separate activities:

Functional Verification: The verification process began with running a demonstration script for the complete run of the clinical workflow. The measure of success for this category of evaluation was that every step (upload, encrypt, store, retrieve, decrypt, validate the integrity of the data and ultimately share with authorized personnel) had to be completed without error. The completion of this phase demonstrates that every component of the clinical workflow integrated and worked seamlessly as a complete system.

Handling Real-World Data: To evaluate the utility of the proposed framework in a real-world context, this study was performed using a set of real-world medical images that included both DICOM (.dcm) and TIFF (.tif) file formats, as they are common format standards for the healthcare industry and also represent a wide variation in size and structure. This assessment shows if the model is capable of accurately modeling the technical characteristics of a real clinical condition in an actual clinical context, rather than a data model that simply contains some randomly generated sample data.

Quantitative Performance Evaluation: A further study to evaluate the performance of the model has been performed to establish the quantitative performance characteristics of the framework's processes on key operational tasks. The performance characteristics used for this study were as follows: (1) The processing time (latency) of each task; (2) The amount of data transferred between the client and server (throughput); (3) CPU usage of the client machine during each task; (4) The amount of RAM used to encrypt the client's data and also during the upload of client-side checkpoint files; (5) The process of downloading and decrypting those files. The quantitative performance evaluation results provide critical empirical information related to how resource-efficient the proposed framework has been designed; its operational efficiency; resource requirements; potential for growth through scalability, etc. This systematic and rigorous testing methodology ensures that an artifact has been built that is both productively useful and backed by empirical evidence, and that there is a thorough understanding of both how the artifact operates and the attributes that need to be improved based on continuous improvement analyses.

# 4 Design Specification

This Secure, Scalable and Auditable System architecture stores image data within the AWS Cloud by leveraging the features of a Multi-Layered Security Framework to provide Confidentiality, Integrity and Availability for Sensitive Patient Data throughout its entire lifecycle (from creation to final archive/deletion). The Multi-Layered Security Architecture leverages four core AWS service offerings, each of which offers specific security and resiliency benefits to provide a comprehensive platform. The backbone of the Multi-Layered Security Framework is Amazon S3 as the primary storage layer due to its high durability (99.999999999%) and virtually limitless scalability for storing encrypted image files. This layer is further protected through versioning, which prevents accidental deletion and preserves history of modifications. At the core of the security model is Amazon KMS, which provides a hardened, FIPS 140-2 validated environment for managing cryptographic keys used for encryption. Amazon DynamoDB serves as the primary metadata store, functioning as a high-performance repository for indexing and retrieving metadata associated with each image. Amazon CloudWatch provides a centralised, tamper-evident log service that records all system activity, enabling a complete audit trail essential for security forensics and regulatory compliance.

Many components work together to provide security through envelope encryption as the primary mechanism for protecting data at rest. When a medical image is uploaded, the system requests AWS KMS to generate a unique Data Encryption Key (DEK). AWS KMS returns both a plaintext DEK and a DEK encrypted with the Customer Master Key (CMK). The plaintext DEK is used to encrypt the image locally using AES-256, then immediately deleted. Two files now exist: the encrypted image stored in S3 and the

encrypted DEK stored in DynamoDB.

To decode the image, the encrypted DEK is retrieved, sent to KMS for decryption using the CMK, and the resulting plaintext DEK decrypts the image. Because each file has a unique DEK, the CMK is never exposed. Role-Based Access Control (RBAC) enforces least-privilege access across three roles: Admin, Doctor, and Patient. For example, a Patient can only view images associated with their own Patient ID, preventing unauthorized access to others' data.

The framework also provides a secure sharing mechanism using pre-signed URLs with time-limited temporary credentials embedded within the URL. These URLs grant temporary read access to a specific encrypted file in S3 without requiring permanent accounts, reducing credential exposure.

An overall auditing framework enhances accountability and non-repudiation. Every major event is logged, including authentication events, permissions, encryption/decryption actions, file-sharing activity, uploads/downloads, and system errors. Each audit entry includes timestamp, user identity, resource identity (e.g., `image_id`), outcome (success/failure), and contextual details. Logs are sent to Amazon CloudWatch to maintain an immutable unified audit trail suitable for compliance and incident-response workflows.

The information system additionally separates large binary images from their associated metadata. Each patient's encrypted image is stored in Amazon S3, while metadata is stored in DynamoDB within a table named `MedicalImageMetadata`. Each entry includes a unique `image_id` as the partition key and other metadata such as `patient_id`, `original_file_name`, `s3_key`, `base64_encoded_encrypted_dek`, and a checksum (SHA-256 hash of the original unencrypted image). The checksum verifies integrity after decryption. A Global Secondary Index (GSI) on `patient_id` enables fast retrieval of all images belonging to a single patient, improving query performance in clinical workflows.

# 5 Implementation

Python is the programming language utilized to develop this framework, as it programmatically interacts with AWS services via the AWS SDK for Python (Boto3). Python was chosen based on several factors: (1) the maturity of its development ecosystem; (2) the availability of many libraries for cloud development; and (3) the packaging of specialised libraries such as cryptography. The codebase is organised into multiple modules, where each module has been designed with a single responsibility. This modular approach improves maintainability, reusability, and overall software quality.

Key implemented framework modules include:

- **Configuration Manager (`config_manager.py`)** Loads configuration settings such as AWS credentials, region, S3 bucket names, and DynamoDB table names from a `.env` file. Separating configuration from logic enhances security and portability.

- **Key Manager (`key_manager.py`)** Interfaces with AWS KMS to create and retrieve the CMK using a predefined alias. It orchestrates envelope encryption, requesting AWS KMS to generate and decrypt DEKs for every image.

- **Encryptor (`encryptor.py`)** Performs AES-256 encryption and decryption in CBC mode using DEKs supplied by the Key Manager. It also computes SHA-256 checksums to verify data integrity. Implemented using `pycryptodome`.

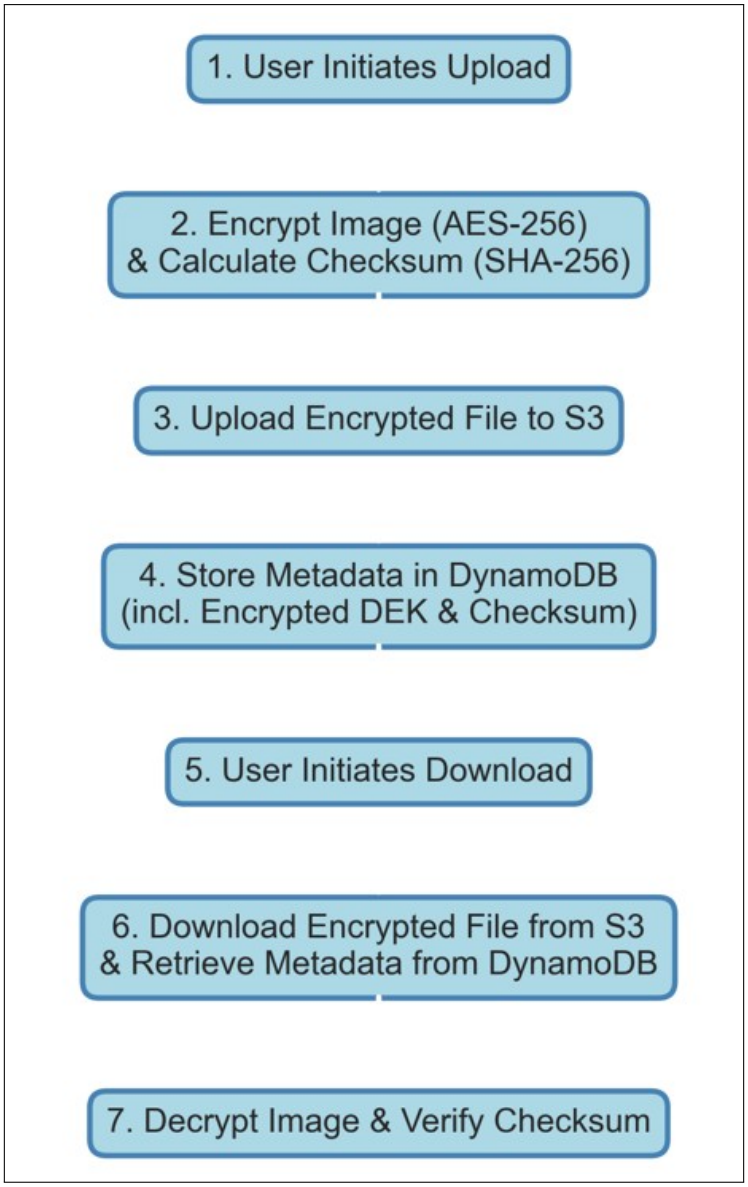


Figure 1: System architecture overview

- **S3 Manager** (`s3_manager.py`) Creates and configures the Amazon S3 bucket programmatically, enabling versioning, server-side encryption, and public access blocks. Provides simplified upload and download functions for encrypted images.
- **Metadata Manager** (`metadata_manager.py`) Stores and retrieves image metadata in DynamoDB. Implements `put_item`, `get_item`, and `query` operations and creates the table with its primary key and GSI.
- **Access Control** (`access_control.py`) Implements Role-Based Access Control (RBAC) with predefined roles (Admin, Doctor, Patient). Enforces permission checks and generates secure, time-limited pre-signed URLs for sharing. Uses an in-memory dictionary for prototype user storage.
- **Audit Logger** (`audit_logger.py`) Provides a central logging facility for all system events. Logs are written both locally and to Amazon CloudWatch for persistent, tamper-evident storage.

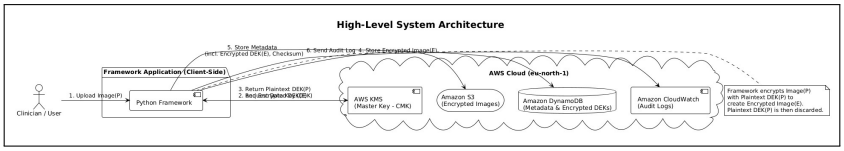


Figure 2: High-level representation of how system components interact.

Additional higher-level scripts developed to demonstrate and test the framework include:

- `setup_infrastructure.py` Automatically provisions all required AWS resources, improving deployment consistency and reducing manual errors.
- `main.py` Serves as the primary entry point for demonstrating an end-to-end workflow, including user creation, image uploads, downloads, and metadata operations.
- `test_medical_images.py` Inputs and processes real-world medical images (DICOM and TIFF formats), validating the system's ability to support clinical imaging data.
- `test_performance.py` Performs bulk upload/download operations and uses the PerformanceAnalyzer module to measure latency, throughput, CPU usage, and memory utilisation.

# 6 Evaluation

Conducted a comprehensive assessment of the developed framework's efficiency in terms of application and real-world effectiveness, through three distinct experiments focused on one of the three capabilities/core functions of the framework. The outcomes of the overall evaluation provide a complete overview as to how viable and effective the framework is for use as a Healthcare-Based System for Data Management for Medical Imaging.

## 6.1 Validation of Functionality and Security Workflow

The purpose of Experiment 1 was to evaluate the usability of the Core Module (functionality) and evaluate whether or not any of the Security Mechanisms (built into the core module) worked as intended. To validate this functionality and security, we created a demonstration script (main.py –demo) that demonstrated one of the standard clinical workflows used in health care. We effectively monitored the console output of the demonstration script and the status of the AWS resources utilized by the demonstration script to determine if each major functionality of the core module was executed correctly. Starting the experiment was successful because all AWS resources were verified and properly configured. These resources included: the medical-images-secure-bucket (S3), the MedicalImageMetadata table (DynamoDB), the Customer Master Key (CMK) (KMS), and the audit log group in (CloudWatch). Then the script created a number of sample user accounts that had varying roles of: administrator, doctor and patient, for the purposes of testing the access control logic.

The primary goal of this experiment was to demonstrate the data lifecycle. A sample X-ray file was created programmatically, and the upload was initiated by a doctor user account. The logs indicated a successful completion of the following processes; first the

access control module checked if the user could write to the bucket, second the encryptor module encrypted the file via client-side using AES-256, computed a SHA-256 checksum for the file, and passed the encrypted file to S3 via the S3 Manager, thirdly the metadata manager completed an entire record in DynamoDB with the following fields: the patient ID, original file name, S3 object key, and the KMS encrypted Data Encryption Key (DEK). Next, the Testing of the download workflow occurred. The doctor user requested to retrieve the same image by using the unique ID. The framework correctly retrieved the metadata from DynamoDB, downloaded the encrypted file from S3, sent the encrypted DEK to KMS for decryption with the plaintext DEK, and used the plaintext DEK to decrypt the image file locally. The most important aspect of this second testing phase is Integrity verification, where the checksum of the newly decrypted file is compared to the original checksum that is stored in the metadata. The results have been validated as being True and completely confirms that the data has not changed and was not corrupted during the entire process.

The last part of the experiment validated the ability to share and retrieve. The doctor user was able to create a secure, time-limited pre-signed URL providing temporary access to the image. In addition, the error to retrieve all images for the specific patient ID returned the correct record, validating that the DynamoDB Global Secondary Indexes are functioning correctly. The first test has verified, without question, that the basic design of the framework along with the various associated features, such as the ones that protect data through encryption, meets all of the required standards and is capable of creating a safe, functional application.

## 6.2 Medical Image File Format Compatibility

The second test has demonstrated that the site is technically capable of supporting a wider range of data types than those provided in the initial test and that it has the ability to validate the reliability of a site's capability to accept and produce standard medical imaging formats, which are typical to those utilized in a clinical environment. To accomplish this, the "test_medical_images.py" script was employed to process the dataset of 12 real-world medical images. The dataset was purposely wide-ranging and included DICOM (.dcm) as well as TIFF (.tif)files, which represent two of the most frequently occurring file types in digital pathology and radiology.

The script worked through all 12 image files in an iterative manner. In each iteration the same sequence of steps was used that was evaluated in the first experiment including a clients' side of encryption, uploading to S3, storing file metadata to DynamoDB, downloading files from S3, and finally decrypting using integrity validation. All 12 images were processed successfully with no errors or failures in the overall workflow. The log files of the script confirm that each file was processed according to the specifications of the encryption and data transfer modules, independent of file format or file size.

At the conclusion of the testing, querying the Framework System statistics relayed the Final State of the framework and provided users with system specifications that indicated a total of 13 images stored within the framework (the 1 demo image plus the 12 images used within Experiment 2) with a total storage amount of 15.02 MB. This data proved that the Framework is not strictly a theoretical construct but actually serves as a viable platform for process and use of the specific types of medical imaging file formats and data structures being used in the medical data industry today. Also, the framework provides evidence that the framework's encryption and storage methodologies are agnostic to the

file format and therefore versatile enough to be used with real-world data.

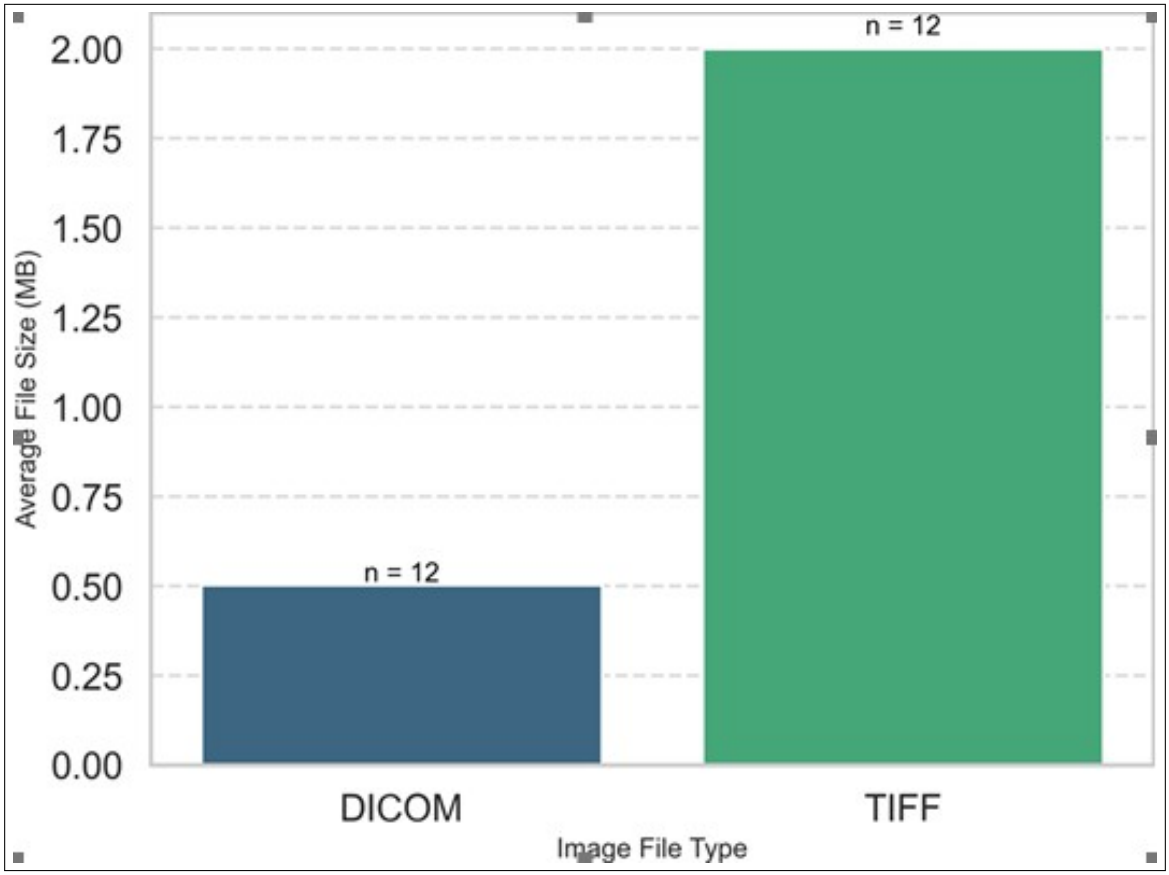


Figure 3: Medical Image Format Processed.

## 6.3 A Quantitative Performance and Resource Assessment

The third experiment was designed to assess the framework's quantitative performance and resource consumption with regards to some of the critical operational activities currently performed by users of the Framework. This experiment provided multiple database measures of the framework's ability to provide empirical values regarding the efficiency, latency measure, throughput, and the client-side electrical and storage consumption of the framework when considering it's practical usage in the resource-sensitive clinical practice environment. The test_performance.py script was executed, allowing the research team to systematically conduct upload, encrypt, download, and decrypt operational tasks using the 12-image medical dataset, generating a total of 24 unique measurements for both upload and download operational assignments.

The mean throughput (total throughput across all 24 transfer operations divided by the total amount of data (30.04 MB)) over the upload and encrypt process was determined to be 0.42 MB/sec. The average upload time per file was determined to be 4.1 seconds; however, the detailed metrics indicated a wide range of durations between the minimum (1.79) and maximum (19.26 seconds). The extended times for the first few operations suggest that one-time latencies (e.g., for establishing secure connections with AWS or other initial client-side setup) are significant factors influencing the longer upload times. As for the subsequent operations, transfer speeds increased significantly and exhibited much greater stability.

The speed and consistency of the download and decrypt process was both faster than the upload process (0.48 MB/sec for download compared to 0.42 MB/sec for upload), and the download process also had less variation (2.5 seconds average operation time with only 0.6 seconds of standard deviation compared to 4.1 seconds average operation time with 1.6 seconds of standard deviation). The higher degree of consistency with the download process is advantageous, because, in order to enable timely clinical decision making, it's important that medical images can be retrieved reliably and quickly.

Throughout the testing of frameworks, client-side resource usage was tracked during performance tests to ensure Frameworks were as efficient as possible with minimal effect on the Client system. Generally, CPU usage during testing was found to be in low single digits, and Memory consumed during testing was steady at the reasonable amount of 71 MB. This means that Client-side Cryptographic Operations were well optimised, allowing for efficient use of both CPU and Memory for the purpose of allowing standard Clinical Equipment to have the framework without the need for high-performance/specialised equipment.

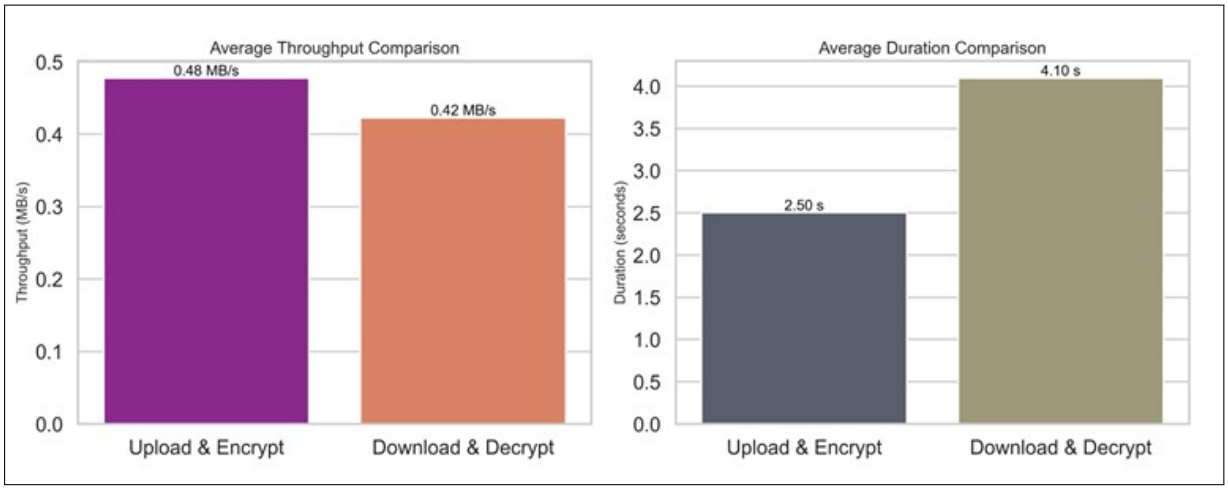


Figure 4: Quantitative Performance

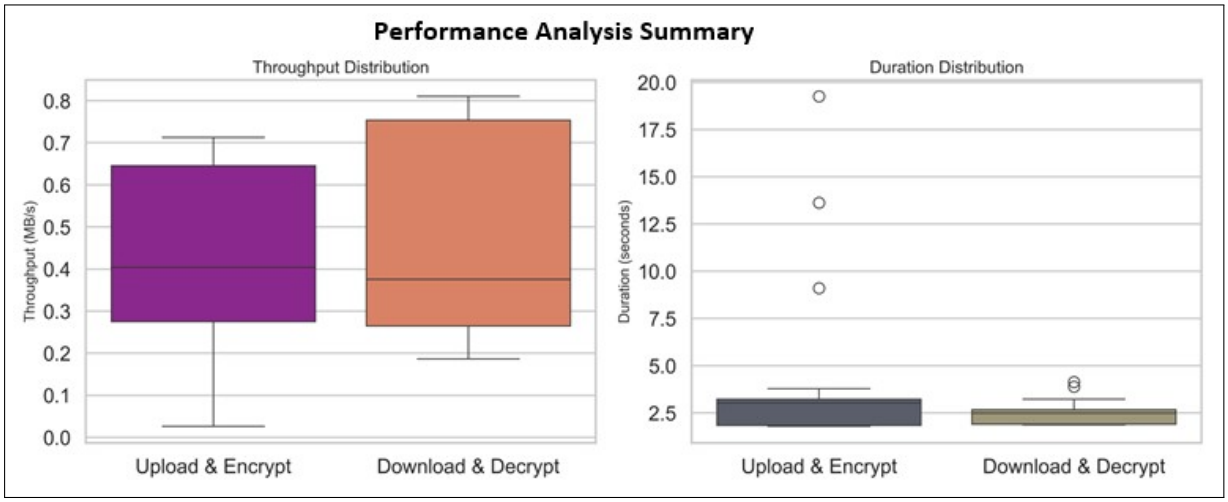


Figure 5: Performance Analysis

## 6.4 Discussion

The results obtained from the three experimental tests compiled, establish that Framework is a secure, strong, robust solution for Cloud-Allied Medical-Image Management, demonstrating that both the functional and practical handling of medical data through Framework (Experiments 1 and 2) confirmed that the Security Architecture effectively operates and operates efficiently and practically with standard Clinical Data Format (CDF).

A detailed performance evaluation (Experiment 3) has quantified the performance characteristics of a system. The throughput rates of 0.4 to 0.5 MB/s are sufficient for a large portion of the types of clinical workflows associated with the retrieval of prior studies for review and/or the asynchronic upload of images associated with diagnostic modalities. With both efficient and consistent download, as well as decryption processes, the findings indicate that rapid and reliable data retrieval in a clinical setting is essential, as quickly diagnosing conditions may often be a matter of life and death.

Notwithstanding the preceding discussion, we must address the limitations of our analyses/assessments. One such limitation noted was that the in-memory user store that was specified for the access control module could serve as a prototypical functionality with which to prototype the concepts discussed; however, a true/persistent/identity management service must function in the production operational realm. Furthermore, the performance benchmarks established were derived from one client at a controlled level. Because a production operation has variable network traffic, multiple clients would be accessing the service, and therefore actual traffic would have an impact on system performance. Even with these limitations, the findings of the comprehensive evaluation confirmed not only the original architecture but also provided substantial evidence to demonstrate that the proposed solution has a high likelihood of successful implementation.

# 7 Conclusion and Future Work

## 7.1 Conclusion

This project solved the challenge of securely managing medical images in the cloud through the design, implementation, and validation of a robust framework. The research showed that by using a set of managed AWS services together KMS for envelope encryption, S3 for durable storage, DynamoDB for indexed metadata, and CloudWatch for auditing an overall, comprehensive, defence-in-depth level of security can be achieved. Validation of the framework confirmed that it meets functional correctness as well as the capability to handle real world medical image formats such as DICOM and TIFF, as well as all aspects of security integrity through checksum validation. The evaluation of framework performance was quantitative and identified that the average upload and encrypt throughput averages 0.42 mb/s, and average download and decrypt throughput averages 0.48 mb/s. This prototype has limitations regarding user management in-memory, yet serves as a viable evidence-based platform on which healthcare organizations interested in adopting "cloud-based" technologies can build upon.

## 7.2 Future Work

This foundational understanding allows us to propose multiple areas for future development that will be beneficial. One of the main goals for achieving production status is connecting the access control module with an enterprise-level Identity Provider (IdP), such as AWS Cognito or a Federated Active Directory. This will support secure and reliable user authentication. In addition, creating a Web-based user interface that facilitates the use of this system by the clinical staff will promote adoption by providing easy access for those who do not have technical skills. Additionally, this framework can be leveraged to support AI and ML models by providing a secure data repository for images that can be analysed using these technologies. Finally, additional research can develop a method for augmenting the existing CloudWatch audit trail by adding a blockchain ledger to allow for decentralised, verifiable and immutable records of data access and sharing. This will significantly increase non-repudiation.

# References


Alabdulatif, A., Thilakarathne, N. and Kalinaki, K. (2023). A novel cloud enabled access control model for preserving the security and privacy of medical big data, *Electronics* **12**(12): 2646.

Ali, M. (2025). A review of secure cloud-based deep learning and chatgpt applications in healthcare, *Global Research Repo* **1**(3): 17–36.

Ali, S. and Others (2024). Cloud security risk assessment, *Cloud Security Review* .

Chauhan, M. and Shiaeles, S. (2023). An analysis of cloud security frameworks, problems and proposed solutions, *Network* **3**(3): 422–450.

Hossain, M. and Others (2024). Iot security principles for connected healthcare devices, *IEEE Internet of Things Journal* .

Kairi, S. and Bhadra, A. (2025). Dna-based cryptography for ultra-secure data, *Advanced Security Computing* .

Kartit, A. (2022). New approach based on homomorphic encryption to secure medical images in cloud computing, *Trends in Sciences* **19**(9): 3970–3970.

Kondylakis, H. and Others (2023). Secure infrastructure for ai-based imaging workflows, *Medical Informatics Europe* .

Lin, X. and Others (2025). Chaotic encryption methods for medical images, *Medical Image Computing* .

Mahajan, D. (2022). Cloud–blockchain ehr architecture, *Journal of Medical Systems* .

Mahammad, S. and Rani, K. (2025). Secure sharing of health records stored in cloud using cryptographic secret sharing schemes through computational intelligence: A review, *Computational Intelligence in Sustainable Computing and Optimization* pp. 281–301.

Montagnese, M., Rangelov, B., Doel, T., Llewellyn, D., Walker, Z., Rittman, T. and Oxtoby, N. (2025). Cloud computing for equitable, data-driven dementia medicine, *The Lancet Digital Health* .

Nazir, S. and Kaleem, M. (2023). Federated learning for medical image analysis with deep neural networks, *Diagnostics* **13**(9): 1532.

Punia, R. and Others (2024). Blockchain-based access control models, *Blockchain in Healthcare Today* .

Shakor, M. and Khaleel, M. (2024). Recent advances in big medical image data analysis through deep learning and cloud computing, *Electronics* **13**(24): 4860.

Theriault-Lauzier, P., Corbin, D., Tastet, O., Langlais, E., Taji, B., Kang, G., Chong, A., So, D., Tang, A., Gichoya, J. and Chandar, S. (2024). A responsible framework for applying artificial intelligence on medical images and signals at the point of care: the pacs-ai platform, *Canadian Journal of Cardiology* **40**(10): 1828–1840.

Vashishth, T., Sharma, V., Sharma, K., Kumar, B., Chaudhary, S. and Panwar, R. (2025). Securing the cloud: Strategies for protecting sensitive patient data in cloud-based healthcare recommender systems, *Healthcare Recommender Systems: Techniques and Recent Developments*, pp. 185–220.

Yan, F., Huang, H., Pedrycz, W. and Hirota, K. (2024). Review of medical image processing using quantum-enabled algorithms, *Artificial Intelligence Review* **57**(11): 300.