International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

Secure Medical Image Sharing With Watermarking In Image Processing

Mr.N.S.R.K Prasad[1],Bangula Srivanth Reddy[2],Errelli Rahul[3],Baddam Siddartha Reddy[4]

[1]Assistant Professor; Department Of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, India.

[2,3,4]B.Tech Students; Department Of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, India.

Mail Id; srivanthreddy53@gmail.com[2]

Abstract

The rapid expansion of internet-based healthcare services and digital medical systems has increased the need for secure handling of medical information, particularly medical images. Protecting patient confidentiality while enabling efficient data sharing among healthcare providers remains a significant challenge. This paper examines various techniques developed to ensure the confidentiality, integrity, and availability of medical data during both transmission and storage. These techniques are broadly categorized into centralized security approaches, including conventional encryption methods and digital watermarking strategies.Special emphasis is placed on the evolution of medical image watermarking, highlighting its importance in enhancing data protection without compromising image quality. Conventional watermarking methods provide advantages such as implementation simplicity, transparency, and compatibility with healthcare regulations. However, each technique involves trade-offs related to computational complexity, scalability, robustness against tampering, and system performance.This study presents a structured classification of existing watermarking approaches and discusses their strengths and limitations. Additionally, potential improvements are identified to enhance security, efficiency, and reliability. The findings contribute to the development of secure frameworks for medical image sharing, supporting trustworthy and collaborative healthcare delivery in increasingly connected medical environments.

Keywords— *Medical Image Security, Digital Watermarking, Data Confidentiality, Encryption Techniques, Healthcare Data Protection, Image Integrity*

INTRODUCTION

Healthcare systems handle highly sensitive information that includes personal patient details, clinical records, diagnostic reports, and medical images. These data elements are essential for accurate diagnosis, treatment planning, and continuous patient monitoring. Any unauthorized modification, disclosure, or loss of such information can result in serious consequences, including incorrect diagnosis, delayed treatment, and compromised patient safety. Therefore, ensuring the confidentiality and integrity of medical data has become a major concern in modern healthcare environments.The rapid adoption of digital technologies, electronic health records, and cloud-based medical platforms has improved accessibility and operational efficiency. However, this digital transformation has also increased exposure to cybersecurity threats. Medical imaging systems, in particular, have become frequent targets of cyber-attacks such as ransomware, phishing, malware injection, and unauthorized access to Picture Archiving and Communication Systems (PACS). These attacks can disrupt healthcare services and compromise patient privacy.One of the most notable incidents highlighting this risk was the WannaCry ransomware attack in 2017, which affected healthcare organizations worldwide. The attack exploited vulnerabilities in outdated operating systems, encrypted critical hospital data, and forced the cancellation of numerous medical procedures. Such incidents emphasize the importance of strong cybersecurity frameworks, regular system updates, and secure data backup mechanisms. In addition, unsecured cloud storage and improperly configured servers have led to exposure of millions of medical images, demonstrating weaknesses in data protection practices.Insider threats also pose serious risks, where authorized personnel may intentionally or unintentionally access confidential patient information without proper authorization. Furthermore, network-based attacks such as Distributed Denial of Service (DDoS) can disrupt access to imaging systems, affecting hospital operations. These threats collectively impact data confidentiality, integrity, and availability, which are essential components of healthcare security.To address these challenges, advanced security techniques including encryption, digital watermarking, artificial intelligence-based threat detection, and blockchain technology are increasingly being explored. These approaches support both centralized and distributed architectures and provide enhanced protection against emerging cyber threats.This study provides an integrated review of medical image security by combining traditional methods such as encryption and watermarking with emerging technologies including federated learning and blockchain. The

proposed analysis bridges the gap between conventional and modern approaches, presenting a unified framework for secure medical image sharing. The major contributions of this work include:

Scope of the Project

The proposed project focuses on the development of a secure cloud-based healthcare management system that integrates Admin, Doctor, Patient, Cloud, and Attacker modules within a single platform. The Admin module supervises system operations, manages user authentication, and approves doctor registrations. Doctors are allowed to register, log in after authorization, manage appointments, review patient requests, and generate digital prescriptions.Patients can register, upload medical reports, schedule appointments, view prescriptions, and receive notifications regarding approvals and updates. The cloud module stores encrypted medical data, ensuring secure access and maintaining confidentiality. An additional attacker module is incorporated to simulate unauthorized access attempts and evaluate the effectiveness of implemented security measures.The system emphasizes secure communication, role-based access control, encryption techniques, and protection against data breaches. Overall, the project aims to enhance healthcare efficiency while ensuring secure handling of medical information.

Objective

The primary objective of this project is to design and implement a secure cloud-based healthcare management system that ensures safe storage and transmission of medical data. The system provides role-based authentication for Admin, Doctor, and Patient users. Doctors can manage appointments, approve patient requests, and generate prescriptions digitally. Patients can upload reports, book appointments, and access prescriptions securely.Another key objective is to store all medical data in encrypted form within the cloud to maintain privacy and confidentiality. The project also includes an attacker simulation module to demonstrate protection against unauthorized access. Overall, the system aims to improve healthcare service efficiency, maintain data integrity, and prevent security breaches.

Problem Statement

Healthcare organizations face significant challenges in securely managing patient records, appointments, and prescriptions. Traditional manual systems and unsecured digital platforms often lead to data loss, unauthorized access, and privacy violations. Sensitive medical data stored without proper encryption is vulnerable to cyber-attacks and tampering.Patients frequently experience difficulties in accessing medical records and booking appointments, while doctors encounter challenges in managing patient data efficiently. Moreover, the absence of integrated cloud-based solutions with strong authentication mechanisms increases the risk of data breaches. Therefore, there is a need for a secure, role-based healthcare system that ensures confidentiality, integrity, and availability of medical data while protecting against simulated attacker threats.

Existing System

Current medical image security systems rely mainly on watermarking and encryption techniques. However, these systems remain vulnerable to various attacks. Active attacks involve intentional modification or removal of watermark information. Copy attacks attempt to duplicate watermark data without authorization. Masking attacks conceal watermark signals, making detection difficult. Forgery attacks introduce fake watermark information, leading to incorrect authentication. Passive attacks focus on extracting watermark-related information without altering the image.

Literature Survey

Several recent studies have focused on enhancing medical image security using encryption and watermarking techniques.Khurana and Rakheja (2024) proposed an asymmetric biometric image encryption method combining Quasi-Zernike synthesis and QR decomposition. The approach improves confidentiality and robustness against noise and compression attacks while ensuring efficient storage and transmission.Baccouri et al. (2024) introduced a lightweight authentication scheme based on elliptic curve ElGamal cryptography. The proposed method enhances secure communication in resource-constrained environments and provides mutual authentication with reduced computational overhead.Dua and Bhogal (2024) developed a medical image encryption technique using a sine-tangent chaotic map. The method demonstrated high entropy, strong randomness, and resistance to differential attacks.Inam et al. (2025) presented a deep learning-based medical image encryption system using CycleGAN networks. The proposed model improves encryption strength while preserving image features and ensuring secure data transmission.Hu et al. (2024) proposed an optical image authentication scheme using computational ghost imaging. The dual-channel encryption approach enhances authentication capability and resistance to common attacks.Khan et al. (2024) introduced a chaotic quantum encryption framework for securing image data. The scheme demonstrated strong resilience against occlusion and differential attacks.Kanwal et al. (2024) developed a blockchain-enabled healthcare image encryption framework using Tinkerbell chaotic mapping. The

system improved data integrity, confidentiality, and resistance to security breaches.

PROJECT DESCRIPTION

This study is organized into multiple sections that collectively address the challenges and solutions related to secure medical image sharing. The paper begins with an introduction to medical imaging security and concludes with a comprehensive discussion of findings and future directions. The central focus of the work is the analysis of medical image watermarking techniques along with secure cloud-based data sharing mechanisms.

The initial sections discuss the evolution of medical imaging, highlighting fundamental concepts, current applications, and emerging challenges. These include the need for scalable architectures, standardized protocols, and improved accessibility in healthcare imaging systems. The study further emphasizes the importance of developing robust security mechanisms capable of protecting sensitive patient data during storage and transmission.Subsequent sections examine both centralized and distributed secure medical image sharing frameworks. These approaches are analyzed in terms of privacy protection, data integrity, and performance efficiency. The review also evaluates the limitations of existing methods and identifies research gaps in secure healthcare data management.The final section focuses on medical image watermarking techniques, including traditional and advanced methods. The effectiveness, limitations, and practical implementation challenges of these techniques are discussed to provide a comprehensive understanding of their role in medical data security.

Methodologies

Medical imaging plays a vital role in modern healthcare by enabling accurate visualization and analysis of anatomical structures. Continuous advancements in imaging technologies have improved diagnostic precision, treatment planning, and patient outcomes. However, these developments also require robust security mechanisms to protect medical data from unauthorized access and tampering.The proposed methodology introduces a secure cloud-based healthcare management system that integrates multiple modules. These modules work collaboratively to ensure confidentiality, integrity, and availability of medical data. The system architecture consists of Admin, Doctor, Patient, Cloud, and Attacker modules.

Technique Used or Algorithm Used

Proposed Algorithm

The proposed system employs a combination of digital watermarking and Advanced Encryption Standard (AES) encryption to enhance medical image security. The watermarking technique embeds essential information such as patient identification within the medical image without affecting visual quality. The image is divided into frequency components, and watermark data is inserted into a sub-band that is less sensitive to image processing operations. This ensures that the watermark remains intact even after compression or minor modifications.To further enhance security, watermark information is encrypted using AES before embedding. AES is a symmetric key encryption algorithm that converts plaintext into ciphertext using a secure secret key. It is widely adopted due to its high security and efficient performance. The overall system integrates multiple modules including Admin, Doctor, Patient, Cloud, and Attacker. Each uploaded medical report undergoes watermark embedding followed by AES encryption before being stored in the cloud. When authorized users request access, the system verifies credentials, extracts the watermark, and decrypts the data. This dual-layer protection ensures confidentiality, authenticity, and integrity of medical images.

REQUIREMENTS ENGINEERING

Advancements in medical imaging technologies have significantly improved healthcare services by enabling accurate diagnosis, treatment planning, and collaborative decision-making. However, these improvements also introduce challenges related to data security, system performance, and interoperability. A clear understanding of system requirements is essential to ensure reliable and secure sharing of medical images. This chapter presents the hardware, software, functional, and non-functional requirements necessary for implementing the proposed secure cloud-based healthcare management system. The objective is to ensure that the system operates efficiently while maintaining confidentiality, integrity, and availability of sensitive medical data. A 15.6-inch LED monitor is sufficient for user interaction, while a minimum of 100 GB hard disk storage is required for application files and database management. Standard input devices such as a keyboard and mouse are necessary for system operation, and an optional webcam may be used for additional authentication features. These hardware components ensure that the application runs effectively during development and deployment.The software requirements specify the development environment and runtime platform necessary for building and executing the system. The proposed system uses Java EE technologies such as JSP and Servlets for front-end development, while MySQL version 5.5 or higher is used for backend database management. The application runs on Windows 10 or Windows 11 operating systems, with Apache Tomcat 7.0 serving as the web server. The system supports commonly used browsers such as Google Chrome and Mozilla

Firefox for user access. Development is carried out using integrated development environments like Eclipse IDE or IntelliJ IDEA. These software components collectively provide a stable and scalable environment for application development, testing, and deployment.Functional requirements describe the operations that the system must perform. The admin module allows the administrator to log into the system using valid credentials and review doctor registration requests stored in the database. The admin can approve or reject registrations and monitor system activities to maintain control over user access. The doctor module enables doctors to register, log in securely, manage appointment requests, approve patients, and generate prescriptions.

DESIGN ENGINEERING

The design engineering phase defines the structural and behavioral representation of the proposed secure healthcare system. The system is developed while preserving key characteristics such as coding efficiency, secure access structure, and strong protection mechanisms. The design focuses on maintaining high performance and ensuring that security features remain intact during data transmission and storage. To accurately represent system functionality and interactions, Unified Modeling Language (UML) diagrams are used. These diagrams provide both static and dynamic views of the system and help in understanding relationships between components, data flow, and operational behavior. The major UML diagrams used in this design include use case, class, object, state chart, sequence, collaboration, activity, component, deployment, and data flow diagrams. Each diagram highlights specific aspects of the system architecture and helps in ensuring structured implementation.

Use Case Diagram

The use case diagram describes the interactions between the primary actors and the healthcare system. The system includes five main actors: Admin, Doctor, Patient, Cloud, and Attacker. The Admin is responsible for managing doctor registrations, approving users, and maintaining the database. The Doctor registers, logs in, approves appointments, manages patients, and uploads prescriptions. The Patient registers, uploads medical reports, books appointments, views prescriptions, and receives notifications. The Cloud actor securely stores patient reports, doctor information, and prescriptions. The Attacker actor represents unauthorized users attempting to access or manipulate data, allowing evaluation of the system's security features. This diagram provides an overall view of system interactions and highlights secure workflows.

Class Diagram

The class diagram represents the static structure of the healthcare system. It includes core classes such as Admin, Doctor, Patient, Cloud, and Attacker. The Admin class contains attributes such as adminId and password, along with methods like login() and approveDoctor(). The Doctor class includes doctorId, name, specialization, and functions such as register(), approveAppointment(), and createPrescription(). The Patient class contains patientId, name, and medicalReport attributes, with methods including uploadReport() and bookAppointment(). The Cloud class manages data storage functions such as storeReports(), storeDoctorDetails(), and storePatientDetails(). The Attacker class simulates unauthorized access attempts. Relationships between classes show that the Admin approves Doctors, Doctors interact with Patients, and the Cloud stores all relevant data.

Object Diagram

The object diagram illustrates real-time instances of system classes and their interactions. Objects such as user, data owner, admin, file, and database are represented with actual values. For example, a user logs in using credentials verified by the system. The data owner uploads encrypted files that contain metadata such as file identifiers and hash values. The admin monitors users and system activities to maintain security. All objects communicate through the database, which stores user details, encrypted files, and verification identifiers. This diagram provides a snapshot of system operations at a particular time.

State Chart Diagram

The state chart diagram describes system behavior by showing transitions between different states. For instance, a patient moves from Registered to Logged In, then Uploading Report, Booking Appointment, and finally Viewing Prescription. Similarly, a doctor transitions from Registered to Approved, then Active, and later Creating Prescription. The admin transitions between Logged Out, Logged In, and Approving Doctor states. These transitions are triggered by user actions or system events. The diagram helps in understanding dynamic behavior and ensures consistent workflow management.

Sequence Diagram

The sequence diagram illustrates the chronological order of interactions between system components. The process begins when a patient logs in and requests an appointment. The doctor receives the request, approves it, and uploads the prescription. The cloud module stores reports and prescriptions securely, while the admin monitors approvals. Messages flow sequentially between components, including authentication, appointment confirmation, report upload, and data storage. This diagram clearly demonstrates the communication pattern among system actors.

Collaboration Diagram

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

The collaboration diagram shows how system objects communicate with each other to complete tasks. The patient sends appointment requests to the doctor. The doctor communicates with the admin for approvals and uploads prescriptions to the cloud. The cloud stores patient and doctor data in the database. Message numbering indicates the sequence of communication. This diagram highlights cooperation among components and efficient data exchange.

Activity Diagram

The activity diagram represents the workflow of the healthcare system. The process begins with user authentication for admin, doctor, and patient. The admin approves doctors, the doctor manages appointments and prescriptions, and the patient uploads reports and books appointments. The cloud stores data securely. Decision nodes represent approvals or rejections. The workflow ends when data is successfully stored and actions are completed. This diagram provides a clear view of control flow.

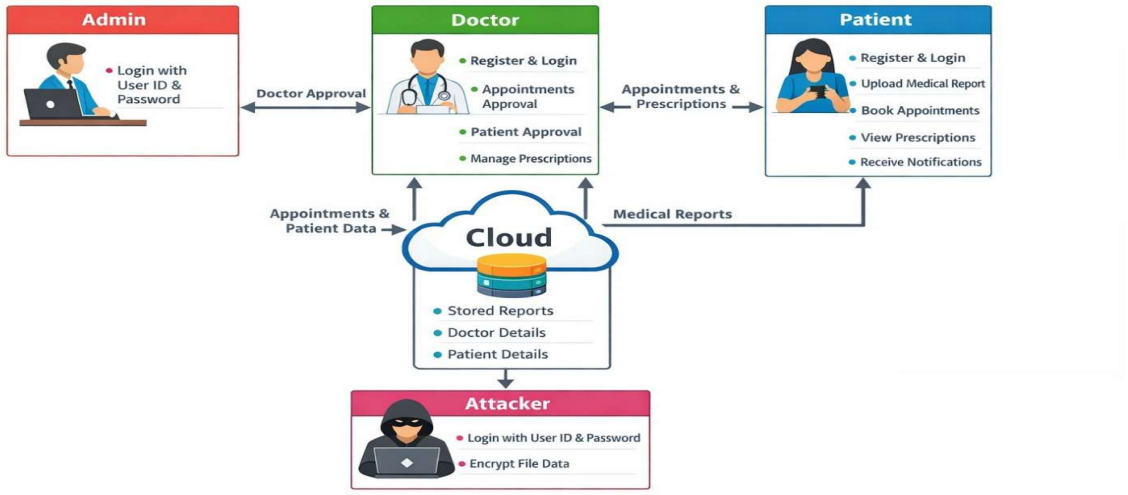
Component Diagram

The component diagram describes system modules and their relationships. The primary components include Admin Module, Doctor Module, Patient Module, Cloud Storage, Database, and Security Module. Each module performs specific functions such as approvals, appointment management, report uploads, and data storage. The security module ensures encryption and access control. The diagram illustrates integration between modules and shows how they work together to provide secure healthcare services.

Deployment Diagram

The deployment diagram shows the physical architecture of the system. The main nodes include Client System, Web Server, Application Server, Database Server, and Cloud Server. Users access the system through client devices. Requests are processed by the web server and application server. The database server stores structured data, while the cloud server manages file storage and backup. This architecture ensures secure and efficient communication.

System Architecture



The proposed healthcare system follows a multi-tier architecture consisting of presentation, application, and data layers. The presentation layer provides interfaces for admin, doctor, patient, and attacker. Users interact through web-based portals with secure login functionality. The application layer contains business logic that processes requests, manages workflows, and validates data. For example, appointment requests are processed and notifications are sent through this layer. The data layer includes database and cloud storage for storing medical records and user information. A dedicated security module handles encryption, authentication, and access control. This architecture ensures scalability, reliability, and protection of sensitive healthcare data while enabling smooth interaction between system components.

DEVELOPMENT TOOLS

This chapter describes the programming languages and development tools used for implementing the proposed system. The application is developed using the Java platform, which provides a reliable and platform-independent environment for building secure web-based applications. The primary technologies utilized include Java, J2EE, and supporting frameworks. Among these, J2EE is selected for implementation due to its capability to

support enterprise-level applications, web services, and database integration. The use of Java-based technologies ensures scalability, portability, and robust performance, making them suitable for secure healthcare data management.

Features of Java

Java is a widely used programming language developed by James Gosling at Sun Microsystems and released in 1995 as part of the Java platform. The language adopts syntax elements from C and C++ while providing a simplified object-oriented model. Java applications are compiled into bytecode, which runs on the Java Virtual Machine, enabling platform independence. This "write once, run anywhere" capability allows applications to execute on different operating systems without modification. Java is designed to be general-purpose, concurrent, class-based, and object-oriented, with minimal implementation dependencies. Its portability, security, and efficiency have made it a popular choice for developing web applications, enterprise systems, and network-based solutions. Java technology is widely used across multiple environments, including desktop systems, mobile devices, and cloud infrastructures, making it suitable for modern distributed applications.

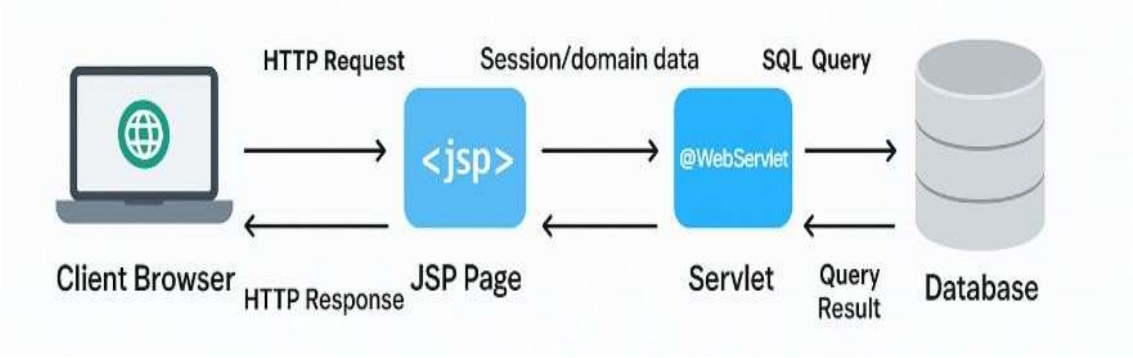
Objectives of Java

Java provides several advantages that attract software developers. It allows programs written on one platform to run on multiple platforms without recompilation. Developers can build web-based applications, browser-based programs, and enterprise-level server-side systems. Java also supports the integration of multiple services, enabling customized application development. Furthermore, it allows development for embedded devices, mobile platforms, and distributed systems.

Advanced Java

Advanced Java technologies are used to develop enterprise and web-based applications. Servlets are server-side programs that process client requests and generate responses. Java Server Pages allow embedding Java code within HTML for dynamic content generation. Java Database Connectivity provides APIs for connecting applications to databases and executing SQL queries. JavaBeans are reusable components that encapsulate data and functionality. The Model-View-Controller architecture separates application logic, user interface, and control flow. Filters are used for request preprocessing, such as authentication and logging. Listeners handle lifecycle events within the application. Session management techniques such as cookies and HTTP sessions track user interactions across multiple pages.

The application is deployed using Apache Tomcat, which acts as a lightweight web server and servlet container. It supports Servlets, JSP, and HTTP protocols, making it suitable for running Java web applications. Tomcat is easy to configure, supports WAR file deployment, and integrates well with development environments. Its lightweight nature makes it preferable compared to full enterprise servers.



SOFTWARE TESTING

Software testing plays a crucial role in ensuring the reliability and quality of the developed system. The primary objective of testing is to identify errors and verify that the application performs according to specified requirements. Testing involves evaluating individual components, integrated modules, and the complete system to confirm that functionality, performance, and security requirements are satisfied. It ensures that the software behaves as expected under normal and exceptional conditions. Various testing techniques are employed to validate different aspects of the system, including functionality, integration, performance, and user acceptance.

Developing Methodologies

The testing process begins with the preparation of a comprehensive test plan that evaluates general functionality and specialized features across multiple platforms. Quality control procedures are applied throughout the testing phase to ensure that the application meets the requirements defined in the system specification document. Testing methodologies focus on identifying defects early, validating system behavior, and confirming compliance with user expectations. The testing framework considers factors such as module-level

verification, integration validation, and system-wide performance assessment.

Types of Tests

Unit Testing

Unit testing is performed to validate the functionality of individual modules. Each component is tested independently to ensure that internal logic operates correctly and produces expected outputs. This testing method focuses on verifying program flow, conditional branches, and input-output relationships. Unit testing is typically conducted after the completion of each module and before integration. It ensures that each part of the system conforms to design specifications and reduces the risk of defects during later stages.

Functional Testing

Functional testing verifies that the application operates according to functional requirements. This testing evaluates whether valid inputs are accepted and invalid inputs are rejected. It also checks that all defined functions perform correctly and generate appropriate outputs. Functional testing ensures that system interfaces, processes, and procedures work as expected. The objective is to confirm that user requirements are fully satisfied.

System Testing

System testing evaluates the complete integrated application. It ensures that all modules function together correctly and meet overall system requirements. This testing verifies configuration settings, workflow processes, and integration points. System testing also examines end-to-end functionality, confirming that the system behaves consistently under real-world conditions.

Performance Testing

Performance testing measures the system's response time and operational efficiency. It evaluates how quickly the system processes requests, retrieves data, and generates outputs. This testing ensures that the application performs within acceptable time limits, even under increased workload conditions. Performance testing helps identify bottlenecks and improves system responsiveness.

APPLICATION FUTURE ENHANCEMENT

Future enhancements of the proposed system focus on improving security, scalability, and reliability. Hybrid methodologies combining technologies such as federated learning and blockchain offer promising solutions for secure medical data sharing. These approaches distribute trust, enhance privacy, and improve data integrity. Additionally, advancements in digital watermarking techniques using artificial intelligence and deep learning can provide stronger protection for medical images. AI-based watermarking methods improve robustness against tampering and maintain image quality. Future research can also explore adaptive encryption mechanisms and intelligent access control systems. Continuous innovation in these areas will help address evolving cybersecurity threats and meet the growing requirements of modern healthcare environments.

Conclusion

This study emphasizes the importance of secure medical data sharing in modern healthcare systems. Various techniques such as encryption, watermarking, blockchain, and federated learning have been analyzed for protecting sensitive medical information. Centralized approaches provide effective confidentiality and integrity but may face challenges in key management and vulnerability to tampering. Decentralized methods offer improved trust distribution and enhanced privacy but require higher computational resources and careful implementation. The comparative analysis highlights the strengths and limitations of each technique. The findings indicate that hybrid approaches combining encryption, watermarking, and intelligent security mechanisms can provide improved protection. Overall, the proposed framework contributes to secure, efficient, and reliable medical image sharing in connected healthcare environments.

References

1) Early research focuses on **digital watermarking techniques** and steganography methods for protecting image integrity and ownership in multimedia and medical data.
2) Several studies discuss **cybersecurity challenges in medical imaging systems**, including PACS security, IoT vulnerabilities, and risks in healthcare data communication.
3) Research highlights the importance of **medical image watermarking and encryption techniques** for maintaining confidentiality and preventing unauthorized access.
4) Various works emphasize **content-based retrieval and AI-driven diagnosis systems**, improving accuracy and efficiency in medical image analysis.
5) Studies explore **advanced medical imaging technologies** such as SPECT/CT, PET/CT, MRI flow imaging, and their impact on modern diagnostics.
6) Multiple references address **privacy protection and secure data sharing** in healthcare systems using encryption and authentication mechanisms.
7) Several authors review **machine learning and deep learning approaches** for medical image denoising, super-resolution, and computer-aided diagnosis.
8) Recent works propose **chaos-based, DNA-based, and elliptic curve encryption**

ISSN 2347–3657
Volume 14, Issue 1, 2026

schemes to enhance security of medical images.

9) Research also investigates **blockchain, federated learning, and hybrid security frameworks** for decentralized and secure healthcare data exchange.
10) Latest studies focus on **AI-driven, lightweight, and post-quantum encryption techniques** to ensure robust, scalable, and future-ready medical data protection.