Serial N^o:

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

ECHAHID HAMMA LAKHDAR UNIVERSITY - EL OUED
FACULTY OF EXACT SCIENCES
COMPUTER SCIENCE DEPARTMENT

Doctoral Thesis

Domain: Mathematics and Computer Science

Field: Computer Science

Specialty: Advanced Information Systems

Title

Security Techniques for Remote Healthcare Systems

By

Mohammed Elhabib Kahla

Composition of the jury:

Mr.	Abdelkamel Ben Ali,	MCA,	University of El Oued,	Chair.
Mr.	Mounir Beggas,	MCA,	University of El Oued,	Supervisor.
Mr.	Abdelkader Laouid,	Pr,	University of El Oued,	Co-supervisor.
Mr.	Ismail Kertiou,	MCA,	University of El Oued,	Examiner.
Mr.	Mohammed Elamine Abderrahim,	Pr,	University of Ouargla,	Examiner.
Mr.	Oussama Aiadi,	MCA,	University of Ouargla,	Examiner.

January 2025

Acknowledgments

First of all, thanks to ALLAH who blessed me with the strength and ability to complete this work.

I am deeply grateful to my Ph.D. supervisor, Dr. Monir Beggas, and co-supervisor, Prof. Abdelkader Laouid, for their unwavering support and guidance from the very beginning. Their expertise and insightful advice have been invaluable in helping me make the right decisions throughout these three long years.

I would also like to thank Dr. Mostafa Kara and Mohammad Hammoudeh for their valuable guidance and support during my research journey.

I am profoundly thankful to all the jury members for graciously agreeing to evaluate my thesis and for their genuine interest in my work.

To all those who have supported me and helped pave the way for completing this doctoral thesis, I offer my deepest thanks.

Lastly, my immeasurable gratitude goes to my family for their unwavering love and support, without which this accomplishment would not have been possible. Special thanks to my parents, my wife, and my siblings for always being by my side.

Mohammed Elhabib Kahla

Abstract

The Internet of Medical Things (IoMT) has revolutionized healthcare by utilizing IoT devices and diverse data sources to provide personalized and efficient care through real-time communication, remote monitoring, and diagnostics. Medical images, central to IoMT systems, require robust security measures to prevent unauthorized access and ensure integrity. However, their sensitive nature, complex networks, and large file sizes increase vulnerability to cyberattacks and unauthorized modifications, which could lead to improper diagnoses and treatments.

This research introduces several innovative advancements to strengthen IoMT security. A novel encryption technique is proposed to reduce processing time and image size while maintaining robust security. A hybrid crypto-system integrates encryption and watermarking to protect medical images efficiently. A lightweight watermarking algorithm based on K-means clustering enhances tamper detection with minimal computational overhead. A blockchain-based watermarking scheme ensures tamper-proof image transmission without third-party involvement. Furthermore, a clustering algorithm is designed to preserve data privacy in resource-constrained IoMT environments, and a dual watermarking framework provides comprehensive tamper detection and authentication.

These contributions collectively enhance IoMT security, reliability, and applicability, supporting the evolution of remote healthcare. The proposed solutions are validated through simulations and real-world scenarios involving network devices.

Keywords: IoT, IoMT, Medical Images, Security, Cryptography, Watermarking, Clustering Algorithm, Blockchain.

Résumé

L'Internet des objets médicaux (IoMT) a révolutionné le secteur de la santé en utilisant des dispositifs IoT et des sources de données variées pour offrir des soins personnalisés et efficaces grâce à une communication en temps réel, une surveillance à distance et des diagnostics avancés. Les images médicales, élément central des systèmes IoMT, nécessitent des mesures de sécurité robustes pour empêcher tout accès non autorisé et garantir leur intégrité. Cependant, leur nature sensible, les réseaux complexes et la grande taille des fichiers augmentent leur vulnérabilité aux cyberattaques et aux modifications non autorisées, pouvant entraîner des diagnostics erronés et des traitements inadéquats.

Cette recherche propose plusieurs avancées innovantes pour renforcer la sécurité des systèmes IoMT. Une nouvelle technique de chiffrement est proposée pour réduire le temps de traitement et la taille des images tout en maintenant un niveau élevé de sécurité. Un système cryptographique hybride intègre le chiffrement et le tatouage numérique pour protéger efficacement les images médicales. Un algorithme de tatouage numérique léger basé sur le regroupement K-means améliore la détection des altérations avec un faible coût computationnel. Une solution basée sur la blockchain garantit la transmission inviolable des images sans intervention de tiers. De plus, un algorithme de regroupement est conçu pour préserver la confidentialité des données dans des environnements IoMT aux ressources limitées, et un cadre de tatouage numérique double offre une détection complète des altérations et une authentification renforcée.

Ces contributions renforcent collectivement la sécurité, la fiabilité et l'applicabilité des systèmes IoMT, soutenant ainsi l'évolution des soins de santé à distance. Les solutions proposées sont validées à travers des simulations et des scénarios réels impliquant des dispositifs connectés.

Mots clés: IoT, IoMT, Images médicales, Sécurité, Cryptographie, Filigrane, Algorithme de regroupement, Blockchain.

ملخص

إن إنترنت الأشياء الطبية (IoMT) قد أحدث ثورة في مجال الرعاية الصحية من خلال استخدام أجهزة إنترنت الأشياء (IoT) ومصادر البيانات المتنوعة لتوفير رعاية صحية شخصية وفعالة عبر التواصل في الوقت الفعلي، المراقبة عن بُعد، والتشخيصات المتقدمة. تُعتبر الصور الطبية عنصرًا أساسيًا في أنظمة IoMT، مما يتطلب إجراءات أمان قوية لمنع الوصول غير المصرح به وضمان سلامة البيانات. ومع ذلك، فإن طبيعتها الحساسة، وتعقيد الشبكات، وحجم الملفات الكبير يزيد من تعرضها للهجمات السيبرانية والتعديلات غير المصرح بها، مما قد يؤدي إلى تشخيصات خاطئة وعلاجات غير مناسبة.

تقدم هذه الدراسة عدة ابتكارات لتعزيز أمان أنظمة .IoMT حيث تم اقتراح تقنية تشفير جديدة تقلل من وقت المعالجة وحجم الصور مع الحفاظ على أمان قوي. كما يتم دمج نظام تشفير هجين يجمع بين التشفير والتوقيع الرقمي لحماية الصور الطبية بكفاءة. علاوة على ذلك، يعتمد خوارزم خفيف للتوقيع الرقمي على تقنية التجميع باستخدام خوارزمية K-means لتحسين اكتشاف التعديلات مع تقليل التكاليف الحاسوبية. كما تم اقتراح حل قائم على تقنية البلوكشين لضمان نقل الصور بشكل غير قابل للتلاعب دون الحاجة إلى تدخل أطراف ثالثة. بالإضافة إلى ذلك، تم تصميم خوارزمية تجميع للحفاظ على خصوصية البيانات في بيئات IoMT ذات الموارد المحدودة، وتم تطوير إطار عمل مزدوج للتوقيع الرقمي لتوفير كشف شامل للتعديلات وضمان المصادقة.

تعمل هذه المساهمات مجتمعة على تعزيز أمان أنظمة IoMT وموثوقيتها وقابليتها للتطبيق، مما يدعم تطوير الرعاية الصحية عن بُعد. وتم التحقق من الحلول المقترحة من خلال عمليات محاكاة وسيناريوهات واقعية باستخدام شبكة أجهزة الحقيقة.

الكلمات المفتاحية: إنترنت الأشياء ، إنترنت الأشياء الطبية ، الصور الطبية، الأمان، التشفير، العلامات المائية، خوارزمية التجميع، البلوكشين.

Contents

List of Figures

List of Tables

General Introduction

Today, one of the most important things is the information that has taken on many different forms, with the development of various technologies and resources, such as devices, sensors, and actuators [1], it has become easy and necessary to create systems composed of numerous interconnected elements. These elements are often geographically dispersed, which leads us to refer to them as IoT systems.

Many definitions of the Internet of Things (IoT) have been proposed in the literature. IoT refers to a network of interconnected physical and digital elements, commonly referred to as "things," that communicate and exchange data over the internet [2]. This definition highlights two key features of IoT systems. The first characteristic is connectivity, where various devices, sensors, and actuators operate autonomously, gathering and transmitting data. These "things" can range from household appliances to industrial machinery. The second characteristic is the seamless integration of these elements into a unified system [3], allowing users to manage the entire network as a single entity, despite the diverse nature of the devices involved.

A prominent application of IoT technology is seen in Remote Healthcare Systems, where IoT-enabled devices monitor patient health in real-time, enabling continuous care outside traditional healthcare settings. This has led to the development of the Internet of Medical Things (IoMT) [4], a specialized branch of IoT focused on connecting medical devices and applications to improve healthcare outcomes. IoMT systems allow for remote monitoring, early diagnosis, and personalized treatment plans, transforming the way healthcare is delivered and making it more accessible, efficient, and patient-centered. However, their sensitive nature, complex networks, and large file sizes increase vulnerability to cyberattacks and unauthorized modifications, which could lead to improper diagnoses and treatments. Thus, robust security measures are essential to prevent unauthorized access and ensure integrity.

IoMT Systems

Historically, healthcare systems have been limited by the need for in-person visits and localized data management, making it challenging to provide continuous

and personalized care. With the advent of IoMT [5], which connects medical devices, applications, and healthcare systems over the internet, healthcare has become more accessible, data-driven, and patient-centric. IoMT enables real-time monitoring, remote diagnostics, and personalized treatment plans, transforming traditional healthcare into a more flexible and responsive system for providers and patients.

IoMT is a rapidly evolving field that integrates IoT technologies with healthcare, creating a network of connected medical devices and systems. Initially focused on basic health monitoring tools, IoMT has expanded significantly with advancements in sensors, data analytics, and cloud computing [4]. Companies like Philips, Medtronic, and GE Healthcare have played a pivotal role in this growth by developing connected devices that support remote monitoring, telemedicine, and data-driven healthcare.

Today, IoMT includes a broad range of applications such as wearable health trackers, smart implants, and connected hospital equipment. According to Deloitte [6], IoMT is defined as "a connected infrastructure of medical devices, software applications, and health systems and services," facilitating advanced healthcare management through data exchange and analysis.

Security

When discussing IoT, it's crucial to highlight the importance of data security, which revolves around ensuring the confidentiality, integrity, and availability of data [7]. These core principles are essential for protecting the various components of IoT systems, including data, software, and connected hardware [8].

1. **Confidentiality**: Ensures that sensitive information is accessible only to authorized individuals [9]. It involves protecting shared resources, preventing unauthorized data extraction, and securing user accounts.
2. **Integrity**: Ensures that data, software, and hardware are protected from unauthorized modifications maintaining their accuracy and reliability.
3. **Availability**: Ensures that systems and data are accessible when needed, even in the event of a security breach, minimizing downtime and maintaining continuous operation.

In an IoMT environment, where interconnected medical devices and systems exchange sensitive data, the security of medical images such as X-rays, MRIs, and CT

scans becomes paramount [10]. Implementing robust security measures is essential to ensure the confidentiality, integrity, and availability of medical images [11]. Here's how each point can be addressed:

- Image confidentiality: Ensuring that only authorized personnel can access medical images.
- Image integrity: Preventing unauthorized alterations to maintain the accuracy of medical images.
- Image availability: Guaranteeing that medical images are accessible when needed by healthcare providers.
- Access control: Implementing secure authentication and authorization to manage access to imaging systems.
- Data segmentation: Isolating medical image data to prevent cross-contamination or unauthorized access within the network.

Principal contributions

The main issue that emerges is security after exploring all this. This leads us to an important question: Does the IoMT provide sufficient protection for transferring medical data and images between all parties ? Our research addresses this question by focusing on the security of medical data transfers. We achieve this by encrypting medical images and employing digital watermarking techniques to secure these images, ensure their authenticity, and detect any potential tampering. Therefore, we have the following contributions:

1. Protecting images is a complex task due to their large size, with both symmetric and asymmetric encryption methods commonly used for this purpose. Asymmetric encryption addresses some of the limitations of symmetric encryption by using a key pair public and private for encryption and decryption, making it harder for attackers to compromise the encryption algorithm. However, this approach also increases encryption time. To overcome these challenges, we propose an encryption technique that utilizes magic number fragmentation to safeguard medical image transfer, aiming to reduce encryption time and image size while maintaining algorithm strength.
2. The limited run-time and computational resources of IoMT devices have driven an increased demand for lightweight, energy-efficient cryptographic systems with robust security primitives to support secure medical image

exchange in IoMT environments. To address these needs, We develop a crypto-system that combines encryption and watermarking to secure medical images within the IoMT environment based on IoT resource limitations.

3. The size of the watermark image, accurate identification of tampered areas, and limited computational resources present significant challenges in protecting medical images. we propose a novel watermarking algorithm designed to detect potential tampering during image transfer. This method employs the K-means clustering algorithm to reduce the watermark image size while ensuring the ability to detect any tampering that occurs.

4. Challenges in watermarking algorithms include constraints on watermark image size and reliance on third parties for the secure transmission of medical images. This has led to a growing demand for an efficient, lightweight spatial watermarking algorithm. To address this, we developed a new blockchain-based watermarking scheme that detects potential tampering and ensures the secure transfer of medical images over a blockchain network without the need for third-party involvement. Our approach aims to balance robust security features with minimal computational overhead.

5. Clustering results often fail to produce data that retains the same characteristics and patterns as the original, which is essential for using it as a substitute in machine learning for ensuring privacy and security. Additionally, there is a need to minimize computational and memory resource usage. To address these challenges, we have developed a new clustering algorithm that preserves the privacy of medical data and is well-suited to the resource-constrained IoMT environment.

6. Most existing watermarking schemes fail to address authentication, copyright protection, tamper detection, and localization simultaneously. To overcome this limitation, we propose a novel dual watermarking algorithm that utilizes both frequency and spatial domains to detect tampering and ensure authenticity, employing a clustering algorithm. This approach is designed for real-time applications and could offer significant advantages for IoT environments.

Thesis Structure

The upcoming Chapter 1 provides an overview of the importance of securing IoMT systems, focusing on the use of cryptography, digital watermarking, and clustering algorithms to safeguard sensitive medical data. In Chapter 2, we examine significant research contributions and approaches for protecting medical images in

IoMT environments. The following chapters expand on this groundwork by introducing proposed techniques and solutions designed to enhance both the security and effectiveness of IoMT systems.

List of publications

Journal papers:

1. An IoMT image crypto-system based on spatial watermarking and asymmetric encryption, Multimedia Tools and Applications, 2024 [12]. **Authors**: Kahla Mohammed El Habib, Beggas Mounir, Laouid Abdelkader, AlShaikh Muath, and Hammoudeh Mohammad.

2. A Nature-Inspired Partial Distance-Based Clustering Algorithm, Journal of Sensor and Actuator Networks, 2024 [13]. **Authors**: Kahla Mohammed El Habib, Beggas Mounir, Laouid Abdelkader, and Hammoudeh Mohammad.

Conference papers:

1. Asymmetric Image Encryption Based on Twin Message Fusion, International Conference on Artificial Intelligence for Cyber Security Systems and Privacy (AI-CSP), 2021 [14]. **Authors**: Kahla Mohammed Elhabib, Beggas Mounir, Laouid Abdelkader, Kara Mostefa, and AlShaikh Muath.

2. K-Means-Based Watermarking Algorithm For medical image, The 7th International Conference on Future Networks and Distributed Systems, 2023 [15]. **Authors**: Mohammed Elhabib Kahla, Monir Beggas, Abdelkader Laouid.

3. Blockchain-Watermarking scheme based K-Means for Medical Image, The 1st International Conference on Smart Mobility and Logistics Ecosystems (SMiLE), September 2024, (accepted). **Authors**: Mohammed El Habib Kahla, Mounir Beggas, Abdelkader Laouida, Mohammad Hammoudeh, Lamri Laouamer.

4. A Novel Dual Watermarking Scheme Based On K-level For Medical Images, The 27th International Conference on Distributed Computer and Communication Networks, September 2024, (accepted). **Authors**: Mohammed El Habib Kahla, Mounir Beggas, Abdelkader Laouida, Brahim Ferik, Mostefa Kara.

Chapter 1

Background

1.1 Introduction

The rapid evolution of technology in healthcare has led to the rise of IoMT, a network of interconnected medical devices, sensors, and software that communicate via the internet. This system has transformed healthcare delivery by facilitating remote patient monitoring, personalized treatments, and more efficient management. However, despite these advancements [16], the widespread adoption of IoMT also brings significant challenges, particularly concerning data security and privacy.

As healthcare systems increasingly depend on IoMT devices to gather, transmit, and process sensitive patient information, securing this data against cyber threats becomes crucial. The distributed nature of IoMT, its varied device types, and the critical nature of the data handled make it a prime target for cyberattacks [17]. Therefore, ensuring the security, integrity, and confidentiality of medical data within this interconnected ecosystem is essential. This chapter explores the security challenges and solutions in IoMT, beginning with an overview of its architecture and applications in healthcare, then addressing key security concerns, and finally introducing core security principles such as cryptography and digital watermarking, which are vital for protecting IoMT environments.

1.2 Overview of Internet of Medical Things (IoMT)

Remote healthcare systems have become integral to modern medical practice, with IoMT playing a crucial role in their operation. IoMT refers to the network of connected devices, sensors, and applications that collect, analyze, and transmit health data over the Internet [16]. This network has revolutionized healthcare by enabling real-time monitoring, data-driven decision-making, and personalized care, all of which rely on various types of big data [17].

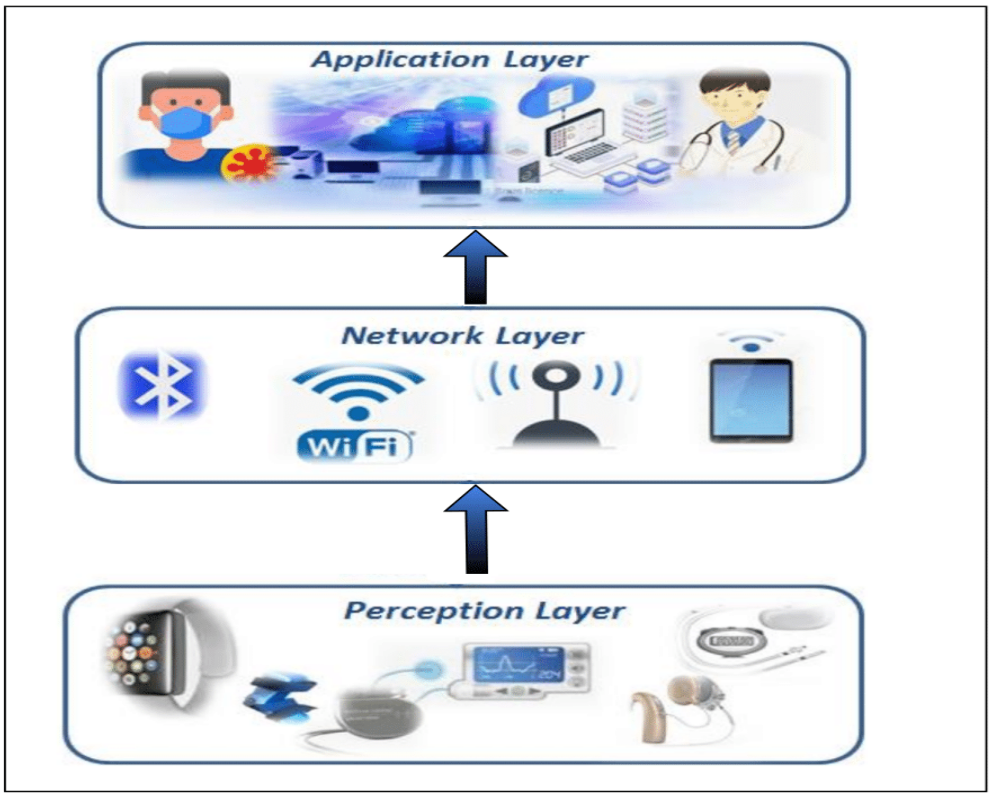


Figure 1.1: Architecture of IoMT.

1.2.1 Architecture of IoMT

The architecture of IoMT is a multi-layered structure that integrates various components and technologies to collect, transmit, process, and store medical data [18]. The architecture is designed to ensure seamless connectivity, data integrity, and security while enabling real-time monitoring and interaction between patients and healthcare providers. The architecture typically consists of the following key layers (see Figure 1.1):

1. **Device Layer (Perception Layer)** encompasses a wide range of connected medical devices, sensors, and wearable technologies designed to collect and monitor physiological data from patients, such as smartwatches, glucose monitors, ECG sensors, and implantable devices like pacemakers [19]. Additionally, some devices in this layer include actuators, such as insulin pumps, which can autonomously respond to the collected data to provide therapeutic interventions.

2. **Communication Layer (Network Layer)** manages the transmission of data from IoMT devices to the cloud or other processing systems, utilizing common communication protocols such as Bluetooth, Wi-Fi, Zigbee, RFID, and cellular networks (4G/5G)

3. **The Application Layer** handles both localized and centralized data processing and storage [19]. and It includes user interfaces like mobile apps, web portals, and healthcare management systems, allowing healthcare professionals, patients, and other stakeholders to interact with the data and devices.

1.2.2 Applications of IoMT

IoMT has significantly expanded the capabilities of remote healthcare, offering numerous applications that enhance patient care, improve healthcare delivery, and optimize medical outcomes. Below are some key applications of IoMT in remote healthcare:

1. **Remote Patient Monitoring (RPM)** uses connected IoMT devices to collect and transmit real-time health data, enabling healthcare providers to manage chronic diseases and adjust treatment plans as needed [20]. It also supports post-surgical monitoring, allowing for the remote detection of complications, and reducing the need for extended hospital stays. This application enhances patient care by providing continuous, data-driven insights.

2. **Telemedicine and Virtual Consultations** powered by IoMT, facilitates virtual consultations by enabling video calls for patient-doctor interactions, allowing for symptom discussion, diagnoses, and prescriptions remotely [21]. It also supports remote diagnosis by transmitting medical images and lab results to specialists for expert analysis without needing in-person visits.

3. **Smart Medication Management** enhances medication adherence and management, particularly for patients with chronic conditions or complex regimens. Applications include smart pills with ingestible sensors that track medication intake and notify healthcare providers, connected inhalers that monitor usage and provide reminders for asthma patients, and automatic dispensers that ensure correct dosage timing and send alerts if doses are missed [22].

4. **Emergency Response and Management** enhances response times in critical situations by providing timely alerts and essential data to healthcare providers [23]. Applications include wearable alert systems, such as

smartwatches with fall detection or heart rate monitors, that automatically notify emergency services during abnormal events like falls or heart attacks, and remote defibrillators that can be activated and located remotely during sudden cardiac arrests, transmitting data to medical teams.

5. **Home Health Care and Assisted Living** aids elderly patients and those with disabilities through remote monitoring and support in home environments [24]. Applications include smart home health systems that use sensors to monitor daily activities and detect irregularities that may signal health issues, and telerehabilitation, where patients recovering from surgeries or injuries can perform physical therapy at home while IoMT devices track their progress and transmit data to therapists for ongoing evaluation.

1.3 Security Concerns in IoMT

This section represents significant IoMT security challenges due to the sensitive nature of the data it handles and the critical role it plays in patient care [25]. Key security concerns include:

1. **Data Privacy and Confidentiality** IoMT devices collect and transmit sensitive health information, making them prime targets for cyberattacks [26]. Unauthorized access to this data can lead to breaches of patient privacy, identity theft, and misuse of personal information.

2. **Data Integrity** The accuracy and reliability of medical data are crucial for effective treatment [27]. Cyberattacks that manipulate or alter data can lead to incorrect diagnoses, inappropriate treatments, and potentially life-threatening situations for patients.

3. **Device Security** Many IoMT devices have limited processing power and memory, making it challenging to implement robust security measures. These devices are often vulnerable to hacking, malware, and unauthorized access, which can disrupt their functionality or cause them to malfunction.

4. **Network Security** IoMT devices rely on wireless communication networks (e.g., Wi-Fi, Bluetooth, cellular) to transmit data [28], which can be susceptible to eavesdropping, man-in-the-middle attacks, and other forms of interception that compromise data security.

5. **Interoperability Issues** The integration of various IoMT devices from different manufacturers can create security gaps if the devices do not adhere to the same security standards [28], leading to vulnerabilities in the overall system.

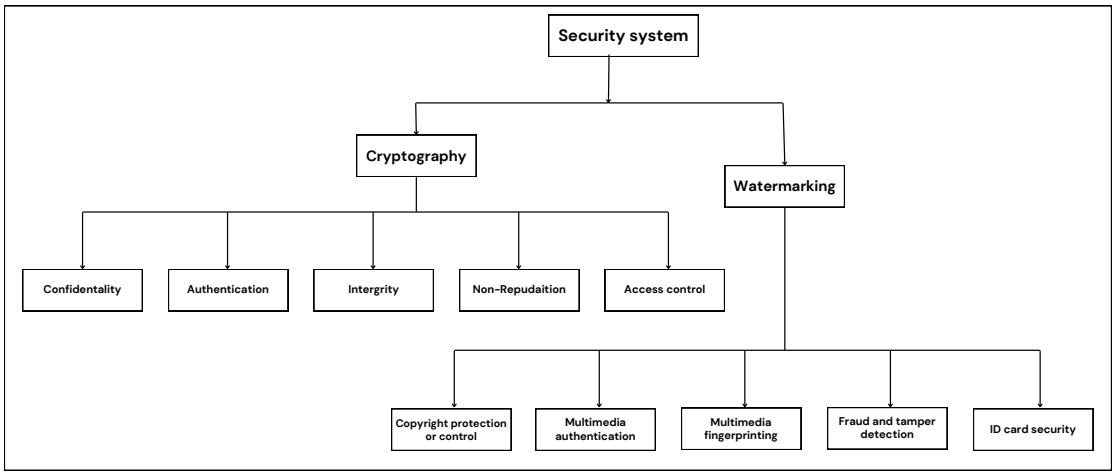


Figure 1.2: Classification of main security system techniques.

6. **Physical Security** IoMT devices, especially wearable or implantable ones, are prone to physical tampering or theft, which can lead to data breaches or unauthorized access to the device's functionality [26].

1.4 Fundamental Security Principles

This section overviews the foundational security principles necessary for safeguarding IoMT environments. Given the critical and sensitive nature of healthcare data, security mechanisms must be robust, reliable, and capable of addressing various vulnerabilities. Two fundamental techniques—watermarking and cryptography—are instrumental in protecting IoMT systems. These techniques have different applications but are both essential for ensuring data confidentiality, integrity, authentication, and overall system security [29] (see Figure 1.2).

1.4.1 Cryptography

The term "cryptography" originates from the Greek words "kruptos," meaning hidden, and "graphein," meaning to write [30]. Essentially, cryptography is the practice of concealing information, whether during communication or storage. Historically, it was predominantly used in the military for securely exchanging sensitive information, such as strategies and plans. However, with the advent of computers in the 1940s and the rise of the Internet in the 1990s, cryptography has expanded into nearly every aspect of public life [31]. Today, cryptography is embedded in everyday items like bank cards and smartphones and is integral to many

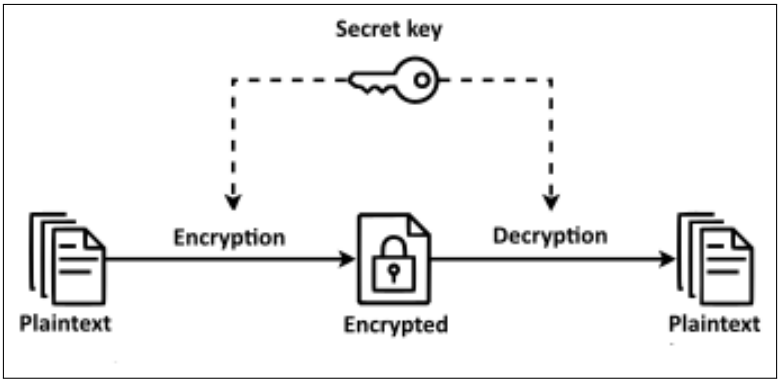


Figure 1.3: Overview of symmetric encryption

computer operations, including secure website access, database management, and online transactions.

Cryptography today is essential for ensuring the data integrity, confidentiality, authentication, and non-repudiation of medical information. This technology is particularly crucial within IoMT's distributed systems and cloud computing environments [32]. It is part of the broader field of cryptology, which also includes cryptanalysis—the analysis of cryptographic systems to identify and exploit vulnerabilities, ensuring the security of sensitive health data against unauthorized access and breaches.

1.4.1.1 Symmetric Encryption

Symmetric encryption, often called secret key encryption, involves the sender and receiver using the same key for encryption and decryption [33]. a symmetric encryption scheme has two algorithms: the encryption algorithm E and the decryption algorithm D. Given a message m and a key k, the ciphertext c is generated as(see Figure 1.3):

$$c = E(m; k) = E_k(m)$$

To retrieve the plaintext from the ciphertext c, the decryption algorithm is applied using the same key k:

$$m = D(c; k) = D_k(c)$$

where the secret key is shared between the communicating parties. The decryption algorithm D is deterministic, meaning it consistently maps a ciphertext-key pair $(c; k)$ to a single plaintext m. In contrast, the encryption algorithm E does not have to be deterministic, in cases where it is randomized, it may produce different ciphertexts for the same message-key pair, leading to ciphertext expansion.

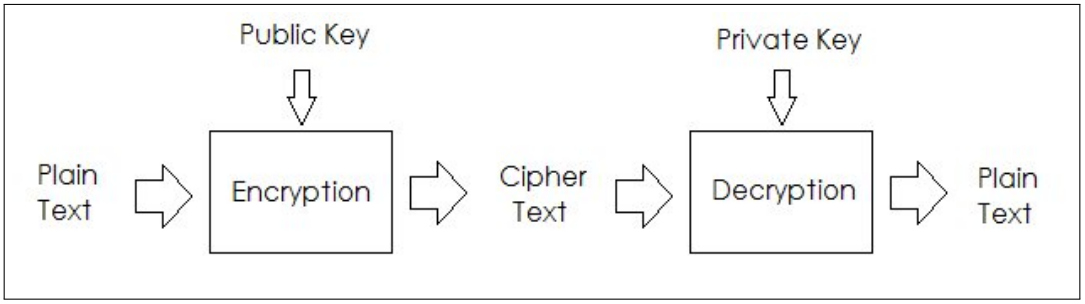


Figure 1.4: Overview of asymmetric encryption

One major fundamental drawback of symmetric encryption is the necessity of a secure channel for sharing the encryption key. Since both encryption and decryption processes depend on the same key, any compromise of this key jeopardizes the security of all communications using it [34]. This requirement for secure key exchange underscores the vulnerability of symmetric encryption in environments where secure channels are not assured.

1.4.1.2 Asymmetric Encryption

Asymmetric encryption, also known as public-key cryptography, uses a public key for encryption and a private key for decryption. Introduced by the Diffie-Hellman key exchange [30], this method features an asymmetric relationship where the public key is widely distributed and the private key remains confidential. It relies on one-way functions that are easy to compute for encryption but difficult to reverse without the private key (see Figure 1.4).

Each user possesses a key pair: a public key pk and a private key sk in asymmetric encryption. These keys are mathematically related, and data encrypted with the public key can only be decrypted with the corresponding private key. To maintain the security of this system, the private key must remain secret.

During the initial exchange, the private key generates a digital signature, enabling other users to verify the key owner's identity. Asymmetric encryption ensures secure communication, as anyone can encrypt data using the public key, but only the holder of the corresponding private key can decrypt it [33]. However, asymmetric encryption is significantly slower than symmetric encryption, often by a factor of up to a thousand times, which is a major limitation in its application.

1.4.1.3 Enhancing IoMT Security Through Cryptography

This subsection explores the critical roles that cryptographic techniques play in bolstering IoMT security across five key aspects, each aspect is vital for maintaining trust and compliance in digital healthcare environments [35].

1. **Confidentiality:** Encryption protects sensitive medical data from unauthorized access, ensuring that only authorized parties can view patient information whether it's stored or transmitted.
2. **Authentication:** Cryptography authenticates users and devices, utilizing digital signatures and certificates to verify that only legitimate entities engage in data exchange within IoMT systems.
3. **Integrity:** Cryptographic hash functions help maintain the accuracy of medical data, preventing unauthorized alterations to ensure that data transmitted or stored remains unchanged and reliable.
4. **Access Contro:** Cryptographic keys and protocols enforce robust access controls, restricting data access to authorized healthcare personnel only, thus safeguarding patient privacy and data security.
5. **Non-Repudiation:** Digital signatures provide a secure and verifiable means of tracking who performed what actions, ensuring accountability and preventing denial of activities in medical transactions.

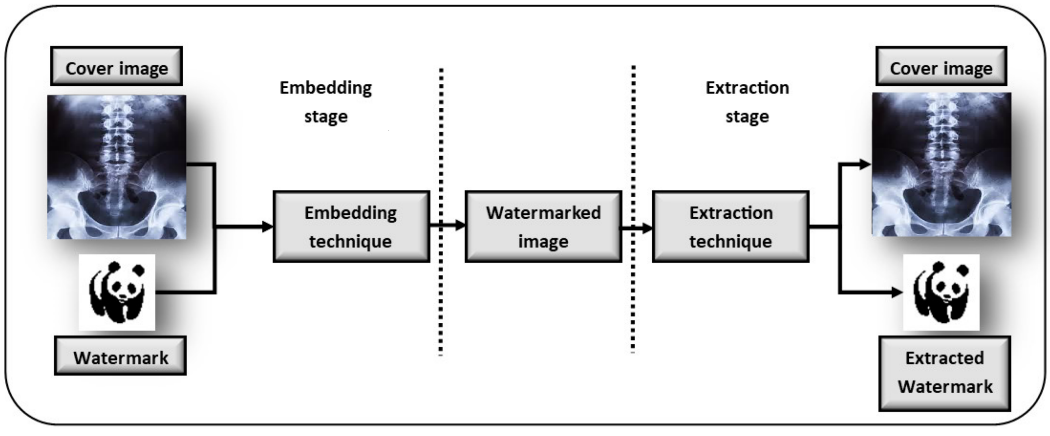


Figure 1.5: Overview of Digital watermarking

1.4.2 Digital watermarking

Digital watermarking emerged in response to the rise of digital media in the late 20th century, adapting the ancient practice of watermarking paper used since the 13th century. This modern technique embeds information into digital media such as images, videos, and audio files to protect copyrights, authenticate content, and trace unauthorized copies [36]. It allows copyright holders to invisibly mark their work in a way that is difficult to remove without degrading the original content's quality (see Figure 1.5).

Digital watermarking today is pivotal for securing medical data and ensuring its integrity, confidentiality, and authenticity. By embedding imperceptible marks into medical images and records, watermarking helps protect sensitive information from unauthorized access and tampering, crucial for maintaining patient privacy and compliance with health regulations. This technology enhances the trustworthiness and security of medical data across various platforms and devices within IoMT ecosystems.

1.4.2.1 Spatial Domain

Those Methods involve embedding a watermark directly into the pixel values of an image or video frame, making subtle modifications that are typically invisible to the naked eye. One common technique within this category is Least Significant Bit (LSB) insertion [37]. In LSB insertion, the watermark is embedded by altering the least significant bits of the pixel values in an image or a frame. This method can modify one or more bits per pixel, depending on the required robustness and the tolerance for image quality degradation. The changes are so minor that they do not perceptibly affect the appearance of the image or video, yet they allow for the watermark to be recovered and verified. This method is prized for its simplicity and low computational cost, making it widely used in applications where real-time watermarking is needed.

1.4.2.2 Frequency Domain

Frequency Domain Methods for digital watermarking leverage transformations such as the Discrete Cosine Transform (DCT) or the Discrete Wavelet Transform (DWT) to alter the frequency components of media [38]. By embedding watermarks into the transformed coefficients, these techniques enhance robustness against compression and other processing effects. The robustness of frequency domain-based watermarking algorithms is often superior to those operating in the spatial domain [39]. Watermarking techniques within the frequency domain are

crafted to ensure that watermarks endure extensive material modifications and resist geometric and non-geometric attacks. In contrast, fragile watermarking techniques found in the spatial domain focus on safeguarding the authenticity and integrity of content.

1.4.2.3 Hybrid Techniques

Hybrid Techniques in digital watermarking combine elements of both spatial and frequency domain methods to achieve a balance between visibility and resilience. By integrating the strengths of both approaches, hybrid techniques can embed watermarks in a way that they are less perceptible in the visual domain while still being robust enough to withstand various types of attacks and manipulations [40]. This dual approach allows for greater flexibility in watermarking applications, optimizing the trade-off between maintaining the original media quality and ensuring that the watermark remains intact under different conditions.

1.4.2.4 Enhancing IoMT Security Through Digital Watermarking

This subsection explores the pivotal role of digital watermarking in maintaining the security and integrity of digital health information within IoMT. The key applications are as follows [41]:

1. **Copyright Protection and Control:** Watermarking secures medical images and documents by embedding ownership and origin data, which is vital for safeguarding intellectual property in medical research and clinical practice.
2. **Multimedia Authentication:** This ensures that medical multimedia content, such as diagnostic scans and procedural recordings, is authenticated, confirming its integrity and reliability for clinical use.
3. **Multimedia Fingerprinting:** Watermarking facilitates the tracking and management of medical media distribution, assisting healthcare providers in controlling access to sensitive information.
4. **ID Card Security:** In IoMT, watermarking is employed to enhance the security of digital ID cards for healthcare professionals and patients, embedding secure, invisible marks to deter forgery.
5. **Fraud and Tamper Detection:** Watermarks play a critical role in identifying unauthorized alterations in medical records and reports, supporting the prevention of medical fraud, and ensuring patient safety.

These applications highlight the critical importance of watermarking in maintaining the security, integrity, and trustworthiness of data across IoMT environments.

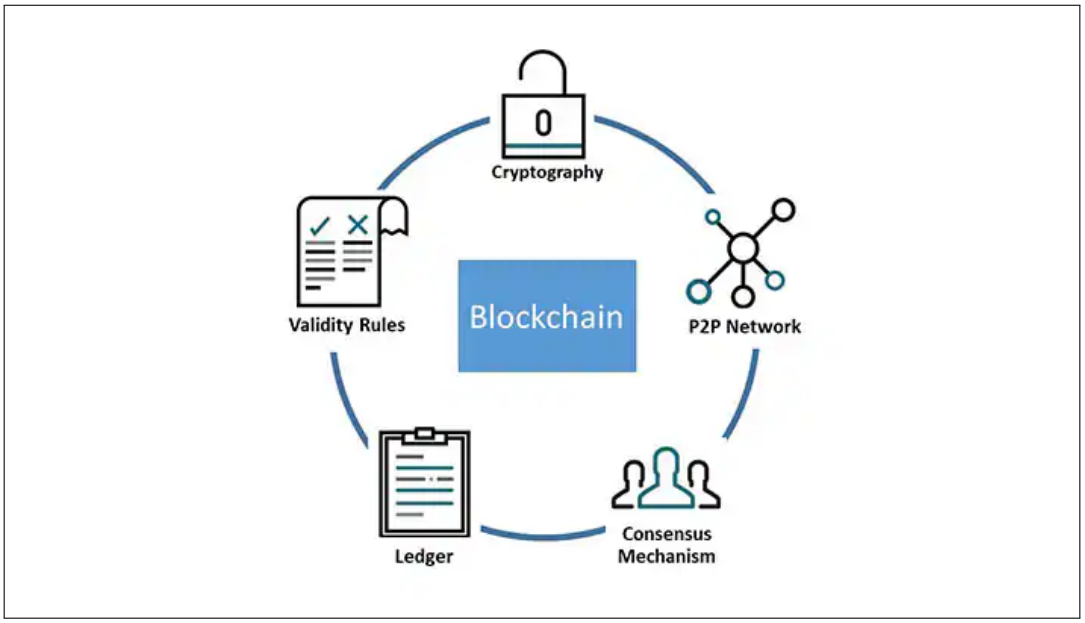


Figure 1.6: Overview of Blockchain Components

1.4.3 Blockchain

Blockchain is an emerging technology that facilitates the storage and transmission of information without centralized control, utilizing peer-to-peer networks to create replicated and distributed ledgers. These ledgers are secured through cryptographic methods, which link blocks together in a chain. Functioning as a distributed database within a network of nodes, blockchain records transactions that are verified and approved before being added to the chain. The transaction details are then stored on a public ledger that is accessible to anyone on the network [42]. The four characteristics of the blockchain are as follows:

- **Powerful Features:** Blockchain provides numerous advantages, including security, stability, transparency, distribution, anonymity, confidentiality, efficiency, and traceability. There are three types of blockchain: public, private, and hybrid. Public blockchains allow any user to join and participate freely, with consensus power typically tied to resources like wealth or computing power [43]. Private blockchains operate in closed environments, where only authorized users with verified identities can participate [44]. Hybrid blockchains combine aspects of both, offering a balance between public accessibility and private control, with a decentralized consensus mechanism controlled by predefined users.

- **Implementation Levels:** Blockchain can be implemented across four different levels, each serving a distinct role in system design. These include protocols for data and network management, distributed consensus mechanisms, frameworks for autonomous organizations using smart contracts, and application interfaces that interact with the blockchain [45]. Each level plays a critical role in the overall functionality and performance of a blockchain system.
- **Key Components:** A blockchain system is built on five essential components: private and public keys for securing transactions, signed chains for linking blocks, a large community of participants, a peer-to-peer document distribution protocol, and a consensus mechanism. The consensus algorithm is particularly important, as it ensures that all nodes agree on the addition of new blocks, maintaining the integrity and consistency of the blockchain [44] (see Figure 1.6).

1.4.4 Clustering Algorithm

Clustering algorithms are unsupervised machine learning techniques used to group a set of objects so that objects within the same group (or cluster) are more similar to each other than to those in other groups. These algorithms play a crucial role in pattern recognition, data mining, and statistical data analysis by identifying natural groupings within data [46].

Those algorithms can be applied across various domains, including market research, image segmentation, and document organization. It is also an essential tool in IoMT, where it is employed to analyze vast amounts of healthcare data, detect patterns, and improve security. In the context of secure IoMT, clustering algorithms help to group similar data points to identify anomalies, segment patient data, and optimize network traffic, thereby contributing to overall system security and efficiency.

1.4.4.1 Clustering Algorithm Types

There are several types of clustering algorithms, each with its strengths and weaknesses, depending on the nature of the data and the desired outcome [47]. The main types include:

- **Partitioning Methods:** Methods like K-means and K-medoids divide data into a predefined number of clusters by assigning points to the nearest cluster center, optimizing cluster distances iteratively.

- **Hierarchical Methods:** These methods create a tree-like structure of clusters by either merging smaller clusters (agglomerative) or splitting larger ones (divisive), effective for small to medium-sized datasets.
- **Density-Based Methods:** Algorithms like DBSCAN form clusters based on data point density, identifying clusters of arbitrary shapes and handling noise, though they can struggle with varying densities.
- **Model-Based Methods:** These methods, such as Expectation-Maximization (EM) for Gaussian Mixture Models (GMMs), assume data is generated from a mixture of distributions and can automatically determine the number of clusters.
- **Grid-Based Methods:** Algorithms like STING and CLIQUE divide data space into grids and perform clustering on these cells, which is efficient for large datasets but may lose detail with coarse grid resolution.

1.4.4.2 Enhancing IoMT Security Through Clustering Algorithms

This section explains how clustering algorithms can assist in this by categorizing data into meaningful groups, which can then be monitored and analyzed for security purposes. Here are some ways clustering algorithms are applied in secure IoMT [48]:

- **Anomaly Detection:** Clustering algorithms like DBSCAN are used in IoMT networks to identify unusual patterns that could signal security breaches, such as unauthorized access or unusual device behavior.
- **Data Segmentation:** K-means can segment patient data based on attributes like vital signs, improving data management and enhancing privacy by ensuring sensitive information is compartmentalized.
- **Secure Communication Optimization:** Hierarchical clustering groups IoMT devices by communication patterns, optimizing network resources and implementing more secure protocols, reducing the risk of data interception.
- **Intrusion Detection:** Model-based clustering, like Gaussian Mixture Models (GMMs), helps create profiles of normal network behavior, flagging deviations as potential intrusions for timely security responses.
- **Risk Assessment and Management:** Grid-based clustering methods assess and manage risks in IoMT networks, allowing healthcare organizations to prioritize security efforts and ensure compliance with regulations.

1.5 Conclusion

IoMT has revolutionized healthcare by enabling remote monitoring and personalized care, but it also introduces significant security challenges. This section discussed the importance of securing IoMT systems using cryptography, digital watermarking, and clustering algorithms to protect sensitive medical data. By ensuring data confidentiality, integrity, and authenticity, these techniques help safeguard patient information and maintain the reliability of healthcare systems. As IoMT continues to evolve, it is crucial to implement advanced security measures to address emerging threats, ensuring that the benefits of IoMT are realized without compromising patient safety.

In the next chapters, we delve deeper into the specific techniques and methodologies proposed in this thesis to enhance IoMT security. Each of these chapters builds upon the foundational concepts discussed here, providing detailed insights and contributions to the field of IoMT security. Together, they form a comprehensive approach to addressing the challenges of securing medical data in an increasingly connected healthcare environment.

Chapter 2

Related Works

This chapter outlines the Related Work, providing a comprehensive overview of existing research relevant to securing medical images in IoMT environments.

2.1 Introduction

With the rapid growth of IoMT, healthcare has been revolutionized by enabling remote monitoring, real-time diagnostics, and data-driven decision-making. Central to these advancements is the handling of medical images, which play a crucial role in patient diagnosis and treatment. However, the sensitive nature of medical images, such as X-rays, MRIs, and CT scans, demands robust security measures to protect against unauthorized access, tampering, and data breaches.

This section provides a detailed review of existing research focused on securing medical images within IoMT environments. Various cryptographic techniques, digital watermarking methods, blockchain applications, and clustering algorithms have been proposed to enhance the confidentiality, integrity, and availability of medical images. By analyzing the strengths and weaknesses of these approaches, this review aims to highlight the current state of medical image security and identify the gaps that the proposed solutions in this thesis aim to address.

2.2 Cryptographic Techniques in IoMT

This section explores the cryptographic techniques currently employed to secure medical data in IoMT environments. A comparison of symmetric and asymmetric encryption methods is also presented, focusing on their effectiveness in safeguarding medical data. Encryption techniques are typically categorized based on the type of encryption process employed: symmetric encryption [49, 50, 51, 52] and asymmetric encryption [53, 54, 55].

Huang et al. [49] introduced a color image encryption system that utilizes a symmetric, plaintext-related approach based on permutation-diffusion techniques and chaotic mapping. Their encryption method involves a single permutation-diffusion operation. The pixel positions in the encrypted image are determined by the initial key stream values generated from chaotic maps. The value of the first encrypted pixel is influenced by both the original image pixels and the key stream elements, which are derived from three one-dimensional chaotic mappings. The values of subsequent encrypted pixels are dependent on the preceding encrypted pixel.

Feng et al. [50] developed an image encryption scheme that employs an invertible chaotic two-dimensional map. This map is split into two components: a left-line map and a right-line map. During the encryption process, a series of iterations of these maps are applied to the image. The number of iterations is determined by the secret key. For instance, if the secret key is "235," the encryption process would involve two iterations of the left-line map, followed by three iterations of the right-line map, and then five additional iterations of the left-line map. Priya Deshmukh [51] introduced an algorithm based on the Advanced Encryption Standard (AES), which supports key sizes of 128, 192, or 256 bits with a fixed block size of 128 bits. The algorithm includes four byte-oriented transformations: substitute byte, shift row, mix column, and add round key. The decryption process is the reverse of the encryption process and involves inverse operations, such as inverse shift row, inverse substitute byte, add round key, and inverse mix columns. The number of rounds in the encryption process varies depending on the key length used. In their work, Arab et al. [56] proposed an image encryption method that combines the AES algorithm with a chaos sequence. Specifically, they utilized the Arnold chaos sequence to generate the encryption key. The authors [57] introduced two modifications to the standard AES algorithm: first, they replaced the permutable operation with a proposed propagation operation, and second, they substituted the column integration operation with a linear transformation operation.

Asymmetric encryption involves the use of a pair of keys: one key for encryption and a different key for decryption. In [53] jiao et al introduced an image encryption scheme that utilizes asymmetric encryption, combining the generalized Arnold map with the RSA algorithm. In this scheme, the parameters of the generalized Arnold map are generated using the RSA algorithm, and the key stream is produced iteratively. The encryption process begins with an XOR diffusion operation, which alters the pixel value distribution to obscure the image data. This is followed by a cyclic confusion of the image's rows and columns, creating an additional layer of concealment. The process concludes with four diffusion and confusion operations rounds, resulting in the final cipher image.Yadav et al. [54] proposed a color image encryption scheme based on amplitude and phase truncation, performed in the fractional Hartley domain. Encryption begins by decomposing the input color image into its RGB (red, green, and blue) channels.

These channels undergo an affine transformation, where the red channel is treated as the amplitude, while the green and blue channels are used as phase masks. The red channel image is then bonded with the green channel phase mask and transformed using the fractional Hartley transform (FRHT). The reserved portion of the resulting image acts as the first decryption key. The truncated part is then bonded with the blue channel phase mask and subjected to another FRHT. The reserved portion from this step becomes the second decryption key, and the new truncated part forms the encrypted image. A notable drawback of this method is the lengthy execution time due to the numerous transformations involved.

In their [55] work, Mehra et al. proposed an optical asymmetric image encryption scheme that utilizes a gyrator wavelet transform to secure an entire image. The encryption keys are derived from the order of the gyrator transform, the position of frequency bands, and the type and level of the mother wavelet. The encryption process begins by applying the Gyrator Transform (GT) to the input image, followed by the application of the Wavelet Transform (WT) to the GT result. To decrypt the image, the inverse of these transforms is used to recover the original image. However, due to the use of multiple transforms, this method is time-consuming for both encryption and decryption and results in a large image size.

2.3 Digital Watermarking in IoMT

In the realm of medical image security and data protection, watermarking techniques are crucial for maintaining the robustness and integrity of patient information during transfer and storage. This section offers an overview of various watermarking methods designed to protect sensitive patient data. Our study assesses representative watermarking approaches in comparison to the proposed method. By categorizing these techniques into spatial and frequency domains, we conduct a comprehensive analysis that highlights the unique characteristics of each category.

In the context of the spatial domain, Su and Chen [58] developed an innovative watermarking algorithm aimed at copy protection, which functions in the spatial domain by utilizing direct current coefficients and encryption. The approach involves splitting the original watermark into multiple sub-watermarks and encrypting each with MD5 using unique keys. These encrypted binary sub-watermarks are then embedded into different locations within the blue channel of the image. However, it is important to note that this method has a relatively limited embedding capacity. In [59], a color image watermarking scheme in the spatial domain is proposed, where a binary watermark image is embedded into a color host image. The embedding process integrates the watermark across all regions of

the host image. The method utilizes the Simple Image Region Detector (SIRD) to select sub-regions from each block, which are then modified to represent the watermark bit using specific masks. The first mask modulates the blue component to embed the watermark, while a second compensating mask adjusts the red and green color components. During the extraction process, the exact embedding regions are identified, and all image blocks are read to produce four sets of watermarks. These extracted watermarks are then reshaped to reconstruct the original watermark. However, a notable drawback of this approach is its vulnerability to certain attacks, particularly rotation attacks. In [60], the authors proposed a method that employs estimated error expansion to embed confidential data into segmented regions of a cover image. The technique automatically segments the image into object and background regions and creates histograms for each. The watermark is then subtly embedded into both regions, with a lower embedding intensity in the object region to reduce the impact on image quality.

The authors [61] introduced a new spatial domain color image watermarking technique. In this approach, the watermark image is first converted into a binary format. Then, each bit of the watermark is embedded into a 4×4 image block. The extraction process simply reverses the embedding steps to retrieve the watermark from the image. In [62], Kundu and Das presented a technique in which a generated medical record watermark is embedded within the encrypted region of interest (ROI) of a medical image. This method leverages AES encryption and compression techniques to create the watermark from the ROI bits of the cover image. Additionally, the SHA-256 algorithm is employed to calculate the hash of the ROI section, ensuring the integrity and security of the medical image. In [63], the authors introduced a watermarking algorithm that embeds watermark data into the RGB plane of a carrier image using a bit plane histogram technique. This method includes a robust detector for extracting key features from the carrier image. Although the algorithm offers strong invisibility and robustness, it has a limited capacity, making it less suitable for scenarios requiring the embedding of large amounts of data. Moreover, its high computational requirements may limit its practicality in environments with limited computational resources.

in another context frequency domain, the authors in [64] proposed an enhanced watermarking algorithm based on DCT that is effective in resisting geometric distortions and noise attacks. The embedding process is divided into two main phases: preprocessing and embedding. In the preprocessing phase, the watermark undergoes an expansion step, where its size is increased to four times its original size, followed by a scrambling step, where a chaotic sequence generated by a Sine equation is used to scramble the watermark. In the embedding phase, the processed watermark is embedded into the host image. To extract the watermark, the inverse of the embedding process is applied. In [65], the authors developed a robust scheme based on the wavelet packet transform. They determined threshold values using vector information derived from the transformed host image. This

vector information, along with the threshold values, was then used to identify the embedding locations within the transformed image. The work [66] presented a digital image watermarking technique that combines SVD, DWT, and DCT with Kalman filtering. In this approach, SVD is applied to both the input image and the watermark image, which are then divided into three components: horizontal, vertical, and diagonal. The watermark image is embedded by replacing the diagonal component of the original image. After this, DWT is applied to the SVD-based image, and the LH (low-high) component from the DWT image is selected. Finally, DCT is applied to obtain the watermarked image. Due to the use of multiple transforms in the frequency domain, including SVD, DWT, and DCT, this method requires significant execution time and substantially increases the image size.

However, it is important to note that the proposed algorithm is resource-intensive and does not include a thorough security analysis. The scheme integrates a chaos-based encryption technique [67] that utilizes the Arnold transform to enhance security. Objective evaluations have shown that it offers strong resistance to attacks while preserving high levels of invisibility. In [68], the authors introduced a dual watermarking technique that integrates DWT,WHT, and SVD. The process begins by decomposing the "Y" and "Cb" channels of the YCbCr carrier image using DWT, followed by applying SVD to obtain the singular matrix of the selected DWT sub-band. Simultaneously, the watermark image undergoes WHT and then DWT, producing four non-overlapping sub-bands. The singular matrix of the watermark image is then embedded into the carrier image. However, this method is characterized by high time complexity, making it impractical for real-time applications. In [69], Loan et al. introduced a watermarking method that embeds an encrypted and scrambled watermark into the DCT coefficients of the "Y" component in a "YCbCr" carrier image. Although the scheme demonstrates strong resistance to brute force attacks, its complexity makes it less suitable for certain real-time applications.

2.4 Clustering Algorithms in IoMT

Clustering algorithms have become a crucial focus of research, particularly in the context of IoMT systems. The ongoing challenge lies in developing robust algorithms that can effectively address the unique security demands of these systems. This section examines existing clustering techniques, categorizing them into three primary types: hierarchical, partitional, and hybrid clustering. Each category's distinct characteristics are explored to provide a comprehensive understanding of their strengths and limitations. Moreover, the significance of clustering in data-intensive environments is highlighted, with a focus on key applications that demonstrate its practical value and relevance in enhancing IoMT security.

The single linkage (SLINK) algorithm [70] clusters data using a bottom-up approach, where, at each step, it merges two clusters that contain the nearest pair of objects that are not yet in the same cluster. While this method can be effective in certain situations, it has notable drawbacks, particularly its complexity and sensitivity to outliers and noise. Another agglomerative clustering method is the complete linkage (CLINK) algorithm [71], which forms clusters based on the maximum pairwise distance between data points. In CLINK, the distance between two clusters is defined as the greatest distance between any pair of objects, one from each cluster. However, relying solely on the nearest or farthest pairwise distances can often lead to inaccurate cluster representations. Both SLINK and CLINK are susceptible to the influence of noise and outliers because they are highly sensitive to individual data point distances. Outliers that are far from their cluster but close to objects in other clusters can cause premature merging, resulting in elongated and less accurate cluster formations. Fionn and Legendre [72] introduced an agglomerative clustering technique that seeks to minimize within-cluster variance by merging pairs of clusters that cause the smallest increase in the weighted squared distance between their centers. However, this method is prone to outliers, which can compromise the quality of the resulting clusters.

Ward's method [73], also known as the Minimum Variance Method (MVM), aims to minimize the sum of squared errors within each cluster. It calculates the distance between clusters using two approaches, typically employing the Squared Euclidean distance when assessing the distance between clusters and a single data object. However, this method is sensitive to outliers, which can significantly affect variance estimates. Additionally, the need for frequent variance updates throughout the clustering process contributes to its computational complexity. In centroid-linkage clustering [74], the center of each cluster serves as its representative, with the distance between these centers used to determine the distance between clusters. This method streamlines the calculation of inter-cluster distances by concentrating only on the distances between the cluster centers, rather than considering all pairwise combinations. However, if the centers are not accurately chosen, the clustering results may be less effective. Moreover, the need to recalculate the center of the merged cluster at each step increases the time complexity of the algorithm. Another method, called Group Average Linkage or Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [75], builds a hierarchical tree based on a pairwise similarity matrix. UPGMA iteratively merges the closest clusters to create higher-level clusters, making it particularly effective for nested or hierarchical data structures. The process starts by treating each data point as an individual cluster. At each step, the algorithm merges the clusters with the closest average similarity, using the arithmetic mean of the pairwise distances between data points in the merging clusters to calculate the distances between clusters.

Unlike hierarchical clustering algorithms, partitional clustering algorithms divide

the data into flat partitions, optimizing a predefined criterion. Two prominent techniques in this category are K-means [76] and K-medoids [77]. The core concept of K-means involves iteratively calculating and updating the cluster centers based on the mean of the data points until specific convergence criteria are satisfied. K-medoids, an enhancement of K-means tailored for discrete data, select the data point nearest to the center of the cluster as the representative, or medoid, of that cluster. Rezaee and Ramze in [78] introduced Fuzzy C-Means (FCM), an extension of K-Means that permits data points to have varying degrees of membership across multiple clusters. FCM iteratively updates cluster centroids and membership values, guided by a fuzziness parameter (m). Although FCM is advantageous for handling complex datasets, it is sensitive to initial conditions, computationally intensive due to its iterative nature, and may face difficulties in achieving convergence. In [79], the author introduced a clustering method known as Partitioning Around Medoids (PAM), which uses medoids as the centers of clusters. The process begins by selecting initial medoids, followed by assigning data points to the nearest medoids. The medoids are then iteratively updated to reduce intra-cluster distances. Despite its effectiveness, PAM is sensitive to the initial selection of medoids, which can lead to suboptimal results. Additionally, its computational complexity is higher than that of K-means, making it less suitable for large datasets.

A new algorithm called X-Means was introduced by Mughnyanti et al. [80], which enhances K-Means by automatically determining the optimal number of clusters using the Bayesian Information Criterion (BIC). The process begins with a single cluster and progressively explores potential divisions, comparing BIC scores at each step. This recursive approach aims to produce an optimal clustering solution for uncovering data patterns. However, the iterative process can become computationally intensive, particularly when dealing with large datasets. In [81], the authors introduced CLARANS, a clustering algorithm derived from CLARA and based on a randomized search approach. Designed for large datasets, CLARANS selects initial medoids and performs local searches to enhance clusters by swapping medoids. Despite its effectiveness, the algorithm is sensitive to parameter settings and, due to its randomized nature, may not always achieve the global optimum.

Hybrid clustering techniques integrate the strengths of both hierarchical and partitional approaches. In their work, Tran et al. [82] introduced a hybrid data clustering algorithm that blends the Artificial Bee Colony (ABC) method with the K-means algorithm to improve clustering performance. The ABC algorithm, inspired by the foraging behavior of honeybees, is used to determine initial centroids. These centroids are then refined by the K-means algorithm based on the distances between data points. The study by Thang et al. [83] introduces a hybrid data clustering algorithm known as "fastDBSCAN," which builds upon the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm. fastDBSCAN is designed to accelerate the traditional DBSCAN process through a

series of key steps. The algorithm begins with data preprocessing to optimize the input data for clustering. It then conducts an initial clustering phase using the DBSCAN method, followed by a merging process that leverages optimized distance metrics or similarity measures. However, the performance of fastDBSCAN is highly dependent on parameter tuning, which requires a certain level of expertise and experimentation to achieve optimal results. In their work, Kumar et al. [84] introduced a novel hybrid data clustering method that combines the Improved Cat Swarm Optimization (CSO) technique with the K-Harmonic Mean algorithm to enhance data clustering. The approach utilizes CSO, a bio-inspired optimization method, to determine the initial positions of cluster centroids. Subsequently, the K-Harmonic Mean algorithm is applied to refine the cluster assignments by taking into account the harmonic mean of the distances. Lin and Chen [85] proposed the Cohesion-based Self-Merging (CSM) clustering method, which utilizes a cohesiveness metric to guide the merging of clusters. This metric evaluates all data points collectively, enhancing the method's robustness against outliers. The process begins with initial partitioning using K-means, followed by a hierarchical merging of smaller clusters based on their cohesion.

2.5 Blockchain in IoMT

Blockchain technology has gained significant attention in the IoMT environment due to its potential to enhance security, transparency, and data integrity. The decentralized nature of blockchain offers a robust solution for addressing some of the most necessary challenges in IoMT, such as secure data sharing, patient privacy, and the prevention of unauthorized access or tampering. This section reviews the application of blockchain in IoMT environments, focusing on various approaches and their effectiveness in securing medical data.

The adoption of blockchain in IoMT is primarily motivated by the need for a secure and tamper-proof system that can manage and store sensitive medical information. Traditional centralized systems are vulnerable to attacks, data breaches, and single points of failure, which can have severe consequences in a healthcare context. Blockchain's decentralized ledger technology provides a promising alternative, where each transaction is recorded across a distributed network of nodes, ensuring that data remains secure and immutable.

One of the key applications of blockchain in IoMT is secure data transmission and storage. Several studies have proposed blockchain-based frameworks that enable secure communication between IoMT devices, ensuring that data is encrypted and verified before it is stored on the blockchain. For instance, Zhang et al. [86] introduced a blockchain-based framework for secure data sharing in IoMT, where patient data is encrypted and stored on a blockchain, and only authorized parties

can access and decrypt the data using cryptographic keys. In [87], fan et al. proposed a blockchain-based system for managing medical records in IoMT. The system utilizes smart contracts to automate the verification and sharing of medical records among healthcare providers. Smart contracts are self-executing contracts with the terms of the agreement directly written into code, which allows for the automatic enforcement of rules and policies.

Blockchain's potential to enhance patient privacy is another area of interest in IoMT. Liu et al. [88] proposed a blockchain-based privacy-preserving scheme for IoMT, where patient data is anonymized before it is stored on the blockchain. The scheme uses advanced cryptographic techniques, such as zero-knowledge proofs, to ensure that data can be verified without revealing any sensitive information about the patient. Xu et al. [89] proposed a lightweight blockchain framework for IoMT that uses off-chain storage to reduce the amount of data stored on the blockchain.

2.6 Comparative Analysis of Existing Techniques

The comparative analysis of existing techniques is a critical component in understanding the strengths and weaknesses of various methods used in securing IoMT. By evaluating different approaches such as cryptographic techniques, digital watermarking, blockchain integration, and clustering algorithms, this analysis highlights their applicability, efficiency, and limitations within IoMT environments. Such a comparison not only provides insights into the current state of technology but also identifies gaps that can be addressed in future research, ensuring that IoMT systems remain robust, secure, and efficient in handling sensitive medical data.

Technique	Strengths	Weaknesses	Application in IoMT
Symmetric Encryption	- High speed and efficiency - Simplicity in implementation	- Key distribution problem - Vulnerable to key compromise	- Suitable for encrypting large amounts of data in IoMT but requires secure key management
Asymmetric Encryption	- Enhanced security with key pairs - No need for key exchange	- Slower than symmetric encryption - Computationally intensive	- Ideal for securing sensitive data exchanges and authentication in IoMT
Digital Watermarking (Spatial Domain)	- Simple and fast - Direct embedding in the image	- Vulnerable to attacks (e.g., rotation, scaling) - Lower capacity	- Useful for quick and lightweight applications where the security of embedded data is not highly critical
Digital Watermarking (Frequency Domain)	- High robustness against attacks - Better imperceptibility	- High computational cost - Complex to implement	- Suitable for scenarios requiring strong data protection and robustness in IoMT
Blockchain in IoMT	- Decentralized and tamper-proof - Enhances data integrity and transparency	- Scalability issues - High energy consumption	- Effective for secure, transparent, and immutable data management in IoMT but requires optimization for scalability
Clustering Algorithms (Hierarchical)	- Captures nested data structures - Provides a detailed dendrogram	- Sensitive to outliers - High computational complexity	- Useful in hierarchical data analysis and anomaly detection in IoMT
Clustering Algorithms (Partitional)	- Fast and scalable - Easy to implement	- Requires a pre-defined number of clusters - Sensitive to initial conditions	- Suitable for large-scale, real-time data clustering in IoMT with well-defined data patterns
Clustering Algorithms (Hybrid)	- Combines strengths of hierarchical and partitional methods - Flexible and adaptive	- Increased complexity - Requires careful parameter tuning	- Effective in complex IoMT environments where diverse data patterns exist

2.7 Conclusion

In this chapter, we have reviewed the key research contributions and methodologies in securing medical images within IoMT environments, focusing on cryptographic techniques, digital watermarking methods, clustering algorithms, and blockchain applications. The analysis highlighted the strengths and limitations of each approach, emphasizing the importance of balancing security, efficiency, and practicality in medical data protection.

Cryptographic techniques, both symmetric and asymmetric, offer robust security mechanisms, yet they come with trade-offs in terms of speed and computational complexity, particularly in IoMT systems where real-time processing is crucial. Digital watermarking, while providing an additional layer of data integrity and authenticity, varies significantly in robustness depending on whether spatial or frequency domain methods are used. These methods, though effective, must be chosen carefully based on the specific security needs and resource constraints of the IoMT environment.

Clustering algorithms, critical for managing and analyzing the vast amounts of data generated in IoMT systems, are explored in their hierarchical, partitional, and hybrid forms. Each category offers unique benefits and challenges, particularly in dealing with data patterns, outliers, and computational demands. The role of blockchain in IoMT, with its decentralized and tamper-proof features, presents a promising solution for secure data management, though its adoption faces challenges related to scalability and energy consumption.

Overall, this chapter has provided a comprehensive overview of existing techniques and their applications in the IoMT domain. The comparative analysis has identified gaps and areas where current methods fall short, paving the way for further research and innovation. The subsequent chapters will build upon this foundation, introducing the proposed methods and solutions aimed at addressing these challenges and enhancing the security and efficiency of IoMT systems.

Chapter 3

Asymmetric image encryption based on twin message fusion

3.1 Introduction

Effective encryption techniques are crucial for ensuring the security and integrity of sensitive data, particularly in domains where digital assets, such as images [90]., are frequently transmitted and stored. In today's digital age, images are used extensively across various fields, including medical imaging, online education, and advertising. The challenge of maintaining the confidentiality and authenticity of these images has become increasingly significant, especially as they are transmitted over the internet and stored in potentially insecure environments such as the cloud.

To protect sensitive images from unauthorized access and cyber threats, traditional encryption schemes have been employed, with symmetric encryption being one of the most commonly used methods. Symmetric encryption relies on a shared key between communicating parties [91] , offering high data throughput and efficiency. However, it also presents certain drawbacks, such as the need for secure key management and frequent key updates, which can complicate its use in dynamic and insecure environments.

Asymmetric encryption has emerged as a solution to some of these challenges by utilizing a pair of keys—one public and one private—for encryption and decryption [92]. This method enhances security by making it more difficult for attackers to compromise the encryption, though it comes at the cost of increased processing time. Given these considerations, our objective is to develop a novel asymmetric image encryption scheme based on Magic Number Fragmentation and El-Gamal Encryption (MNF-G). This scheme aims to securely encrypt images before transmission, ensuring that their contents remain intact and can be efficiently decrypted. By focusing on reducing encryption time and minimizing the size of the encrypted images, while maintaining strong security, this approach seeks to address the critical need for effective image encryption in today's digital landscape.

3.2 Proposed scheme

Securing and protecting information, particularly images, is a significant challenge today. Images can be safeguarded through various methods, with encryption being one of the most notable. Encryption techniques are generally categorized into symmetric and asymmetric methods. This thesis proposes a novel asymmetric image encryption scheme based on Fully Homomorphic Encryption using Magic Number Fragmentation and El-Gamal Encryption (MNF-G) [93]. Given its common application, this technique is applied to RGB images, and the proposed algorithm is divided into two main phases:

- Encryption Phase
- Decryption Phase

3.2.1 Magic Number Fragmentation and El-Gamal Encryption (MNF-G)

This section explains the MNF-G scheme and its functionality. Consider the following elements: P, C, K, Enc(), Dec(), and the cyclic group $Y_p = Z/pZ = \{0, 1, \ldots, p-1\}$, where:

- P is the ring of plaintext m.
- C is the ring of ciphertext c.
- K is the key ring.
- Enc() is the encryption function $Enc_{pk} : P \longrightarrow C$ with $pk \in K$.
- Dec() is the decryption function $Dec_{sk} : C \longrightarrow P$ with $sk \in K$.

The MNF-G scheme consists of three processes:

1. **KeyGen**: Generates the secret and public keys sk and pk respectively, where $(sk, pk) = (k, k + r \times p, g)$.
2. **Enc(m)**: Encrypts plaintext $m \in P$ using the public key pk, resulting in $c_+ = (m' + m'' \times pk) \mod n$ and $c_\times = (m \times g^m) \mod n$.
3. **Dec(c)**: Decrypts ciphertext using the secret key sk, yielding $m = (c_+ \mod p) \mod (k-1)$ and $m = c_\times \times (g^m)^{-1} \mod p$.

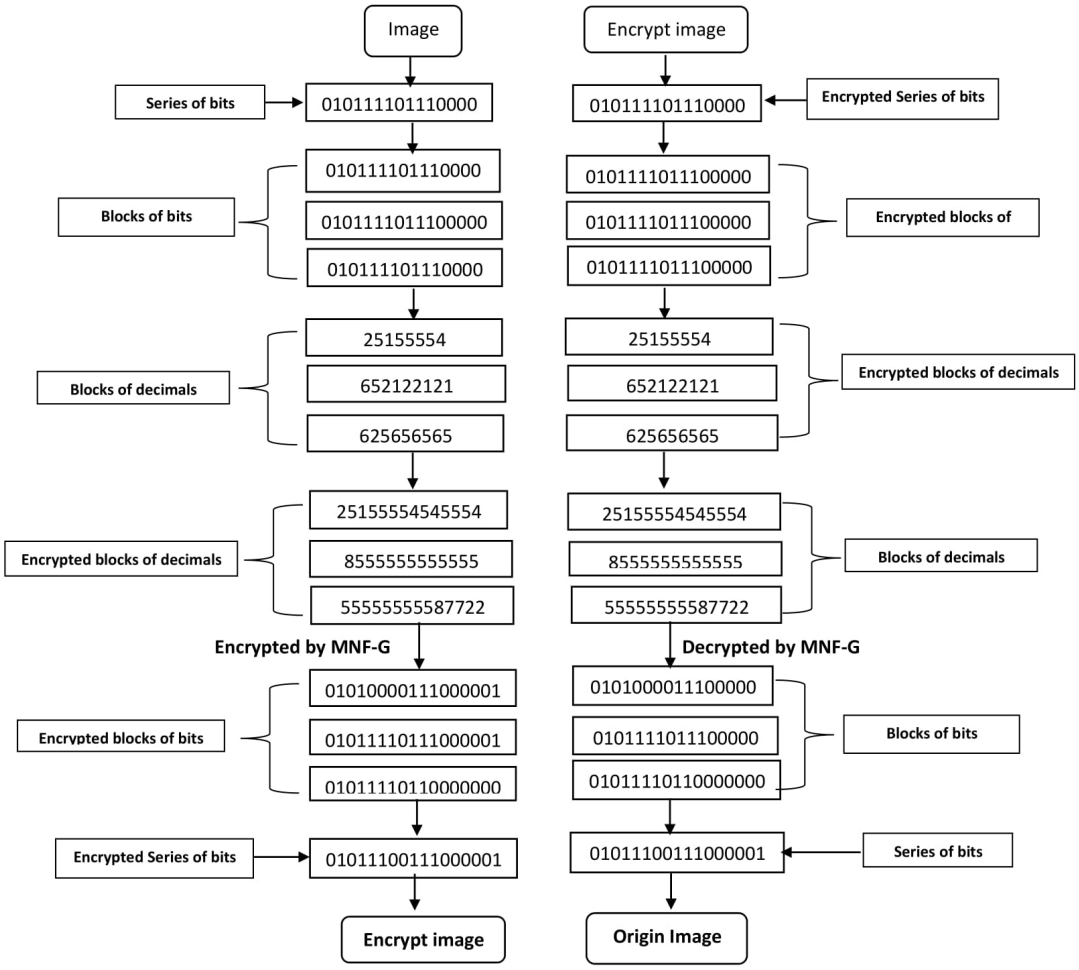


Figure 3.1: Encryption and Decryption Phases.

3.2.1.1 Encryption

The proposed scheme uses the shared public key for encrypting decimal blocks with $pk = k + r \times p$. The encryption process is defined as $c = (m' + m'' \times pk) \mod n$, i.e., $c = m' + m'' \times (k + r \times p) = m' + m'' \times k + r' \times p$. If $m' < p$, then $c \mod p = m' + (m'' \times k) \mod p$. Similarly, if $m'' < p$, then $(m'' \times k \times k^{-1}) \mod p = m''$.

3.2.1.2 Decryption

For decryption, given $c = (c_+, c_\times)$ where $c_+ = m' + m'' \times k + r \times p$, the first step is to extract m from c_+, since $m \times k < p$. Then, $c_+ \mod p = m' + m'' \times k$ because $r \times p \mod p = 0$ and $m \times k \mod (k-1) = m$. Thus, $Dec(c_+) = m$.

This decrypted value is then used to decipher the second part $c_\times$ using g, with $s = Dec(c_+) = m$, resulting in $m = (c_\times \times (g^s)^{-1}) \mod p$.

3.2.2 Encryption Phase

The encryption phase begins by transforming the input image into a series of bits, which are then converted into decimal numbers. These decimal numbers undergo encryption using the MNF-G algorithm. After encryption, the process is reversed: the encrypted numbers are converted back into bit blocks, concatenated into a bit series, and finally transformed into an encrypted image.

3.2.3 Decryption Phase

Figure 3.1 illustrates the decryption process. It begins with converting the image into a series of bits, which are then divided into blocks. Each block is converted to a decimal number to apply MNF-G decryption. The decrypted output (a new decimal number) for each block is converted back into bits. Finally, all bits are concatenated to reconstruct the original decrypted image.

3.3 Implementation

In this section, we detail the implementation of the proposed encryption scheme demonstrate the effectiveness of the encryption and decryption processes, and present the results through visual evidence. This scheme was implemented using Python on a personal computer equipped with an Intel® Core™ i7-8550U CPU @ 1.80GHz and 4GB of RAM, running Windows 10. This setup provides a standard environment for testing the performance and efficiency of the encryption algorithm.

The results of the encryption and decryption processes are depicted in Figure 3.2. The figure contains a series of images showcasing the effectiveness of the encryption algorithm. The encrypted images exhibit no discernible relationship to the original images, demonstrating the algorithm's capability to achieve visual security by effectively obscuring the content. The decrypted images' similarity to the original images confirms the accuracy and reliability of the decryption process.

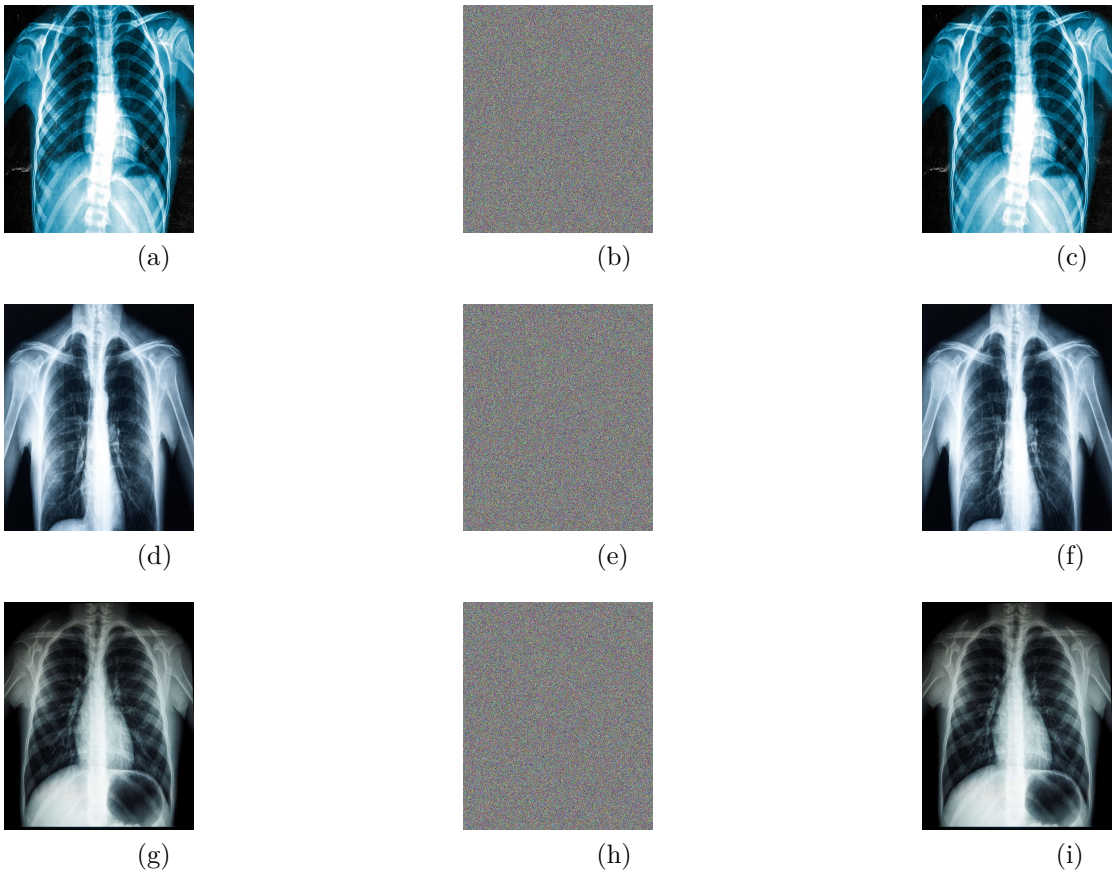


Figure 3.2: The result images of the encryption and decryption phases.

3.4 Evaluation and Results

This section examines the effect and robustness of our encryption scheme through Histogram Analysis and Complexity Analysis. At the end of the section, we compare our scheme with related asymmetric and symmetric schemes, focusing on encryption time, description type, and entropy factor.

3.4.1 Histogram Analysis

A histogram depicts the distribution of pixel intensity in an image. Ideally, an encrypted image should have a uniform frequency distribution, offering no useful information to an attacker.

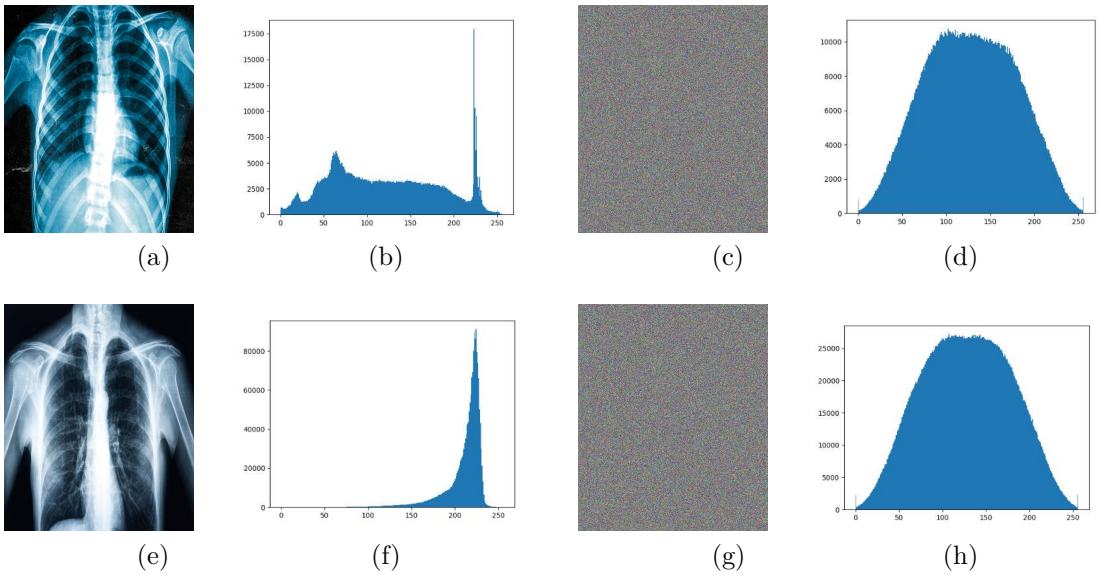


Figure 3.3: The Histograms of images, (3.3a),(3.3e) Origin images, (3.3b) , (3.3f) Histogram of origin images, (3.3c),(3.3g) Encrypted images, (3.3d) , (3.3h) Histogram of encrypted images.

When comparing the histograms of the original images in Figures 3.3b and 3.3f with those of the encrypted images in Figures 3.3d and 3.3h, we observe a clear and significant difference between them. This indicates that decrypting or tampering with the encrypted image is difficult.

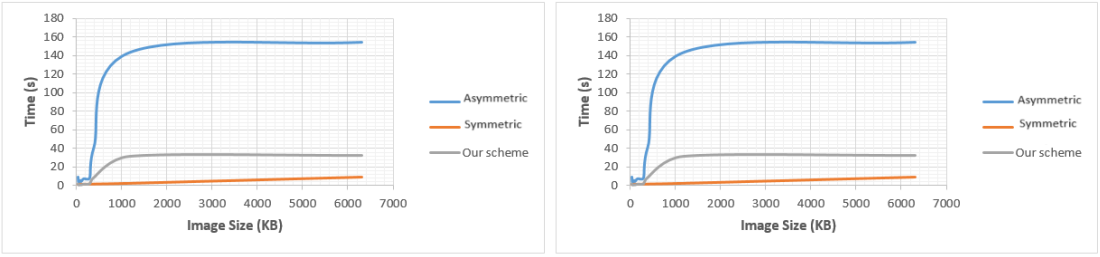
3.4.2 Complexity Analysis

The MNF-G algorithm used is divided into two parts. The first part involves fragmentation, multiplication, and addition operations, resulting in a computational complexity of $O(1)$.

To encrypt the second part, calculating g^m has a complexity of $O(\log(n)^3)$. Decrypting this part, by computing $(g^m)^{-1}$, also results in $O(\log(n)^3)$. Utilizing the vec-power function, the complexity is simplified to $O(1)$ [93].

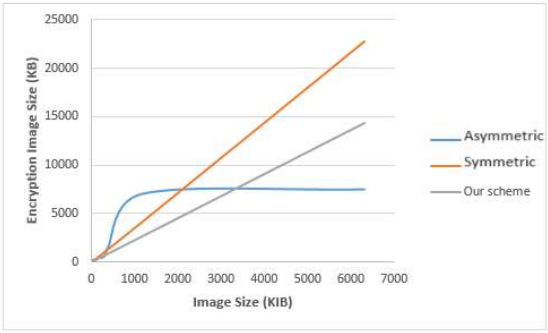
3.4.3 Comparative study

To evaluate our scheme, we compared the time required to encrypt and decrypt image files of varying sizes using two algorithms: one symmetric [52] and one asymmetric [53]. These results, obtained under identical environmental conditions and parameters, are depicted in Figures 3.4a, 3.4b, and 3.4c.



(a)

(b)



(c)

Figure 3.4: Comparative study charts, (3.4a) Encryption image time, (3.4b) Encryption image time, (3.4c) Encryption image size.

Figure 3.4a represents the changes that appear in the time required to encrypt images of different sizes, and Figure 3.4b represents the changes that appear in the time required to decrypt images of different sizes. The size represents the images generated by the encryption, the changes in size images generated by the encryption compared to the input image size shown in Figure 3.4c. Our scheme's encryption time is shorter than that of the asymmetric scheme [53] and comparable to the symmetric scheme [52]. Therefore, our system offers the security strength of asymmetric algorithms while maintaining the speed of symmetric algorithms.

Another way to evaluate our scheme is entropy, which analysis is crucial for evaluating the randomness of information. It is used to measure the unpredictability of data sequences [94]. Entropy is calculated as follows:

$$E(I) = \sum_{n=1}^{256} p(I_n) \log_2 p(I_n)$$

where I represents the image, and $p(I_n)$ is the probability of I_n. The results are shown in Table 3.1.

Table 3.1: Comparison of Entropy's for Three Images

	Image 3.2a	**Image 3.2b**	**Image 3.2c**
[**53**]	7.79595	7.80373	7.797613
[**53**]	7.6282	7.5183	7.802563
Our Scheme	7.82882	7.86239	7.841146

3.5 Conclusion

In this thesis, we introduced an asymmetric image encryption scheme using Twin Message Fusion (TMF) with Magic Number Fragmentation and El-Gamal Encryption (MNF-G). The method proved effective in securely encrypting and decrypting images, ensuring that the original images could be accurately reconstructed without distortion. Our analysis demonstrated that the scheme is robust against various cryptographic attacks, offering a strong balance between security and efficiency. This makes the MNF-G algorithm a promising solution for secure image transmission over the internet.

Chapter 4

An IoMT image crypto-system based on spatial watermarking and asymmetric encryption

4.1 Introduction

Medical images are vital components of IoM systems, playing a crucial role in patient diagnostics and treatment planning. Given the sensitive nature of these images, ensuring their security is paramount, especially as they are transmitted across various networks and devices [95]. The complexity of IoMT environments, with their diverse and interconnected systems, presents significant challenges in maintaining the confidentiality, integrity, and availability of medical images [96]. Unauthorized access or modifications to these images can lead to severe consequences, including misdiagnosis or improper treatment, highlighting the need for robust security measures.

To address these security concerns, techniques such as image watermarking and cryptography have been employed extensively [97]. Integrating these two techniques into a single crypto-system offers a powerful solution for securing medical images in IoMT environments [55]. However, this integration must take into account the limitations of IoT devices, such as limited processing power and energy resources. In this chapter, we propose a dual-layer security system that combines spatial watermarking with asymmetric encryption, specifically utilizing the Twin Message Fusion (TMF) scheme. This approach not only enhances the security of medical images by requiring attackers to bypass both layers but also ensures that the system is efficient enough to operate.

4.2 Proposed technique

The proposed crypto-system for protecting the exchange of digital medical images within an IoMT architecture employs a dual-layer security system. This system includes security and authenticity layers for the medical images. By integrating encryption and watermarking techniques, the content is protected from unauthorized access and any tampering during transmission is prevented. This secure transmission occurs between the perception layer and the application layer via an untrusted third-party network layer, as shown in Figure 4.1.

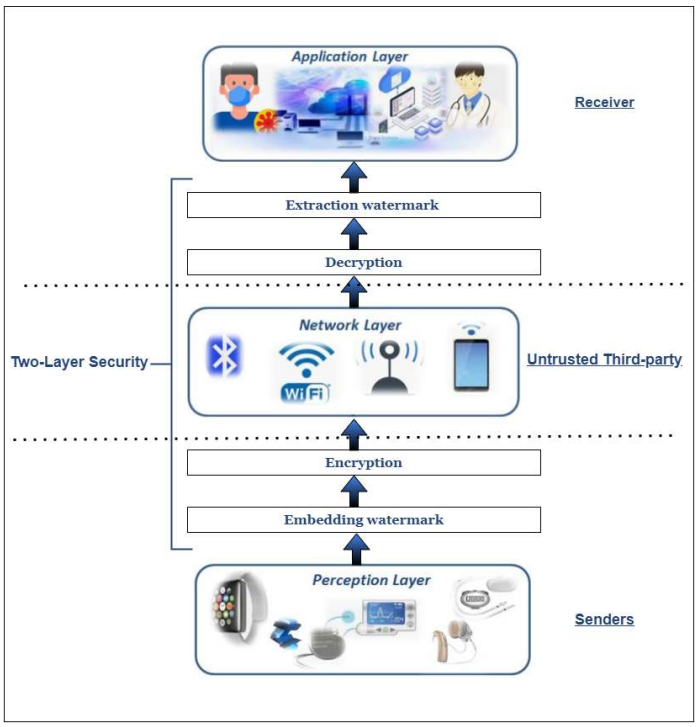


Figure 4.1: Proposed IoMT architecture .

4.2.1 Digital watermarking layer

The watermarking phase protects the medical image by maintaining its integrity and authenticity. Given the constraints of IoT, we propose a new digital watermarking algorithm in the spatial domain, which considers the processing time and the size of the watermarked image. This watermarking process involves two main steps: embedding the watermark and extracting it.

4.2.1.1 Embedding watermark to medical image phase

In the embedding step, the watermark image is converted to grayscale, optimizing both the embedding time and the image size. To enhance resistance to rotation attacks and ensure the distribution of the watermark across the entire image, the embedded watermark, referred to in this technique as the Responsive Watermark (RW), is composed of four sub-watermarks. Each sub-watermark is created by rotating the original watermark image by 0, 90, 180, and 270 degrees, as illustrated in Figure 4.2.

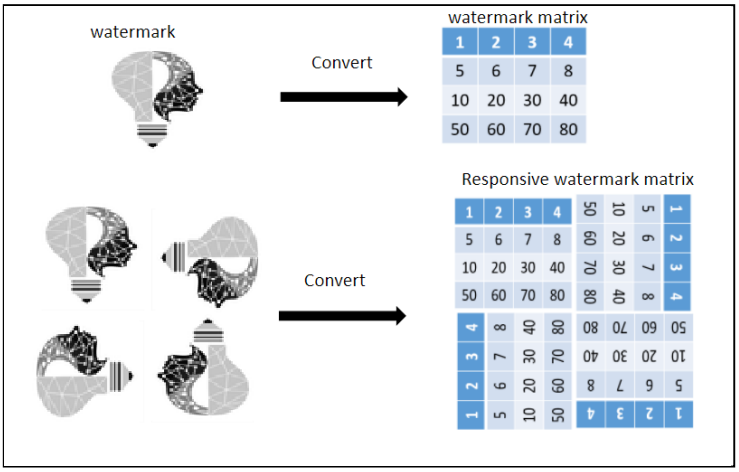


Figure 4.2: Composition of the responsive watermark.

To embed a watermark image into a colored host image, the host image's binary format is represented in RGB, with each pixel consisting of 3 bytes. The embedding process starts by selecting a block of pixels from the center of the host image, matching the watermark's (RW) size, and converting it into a binary format matrix. Each byte in the RW matrix is split into three parts: W1, W2, and W3, representing the first two high bits, the following three bits, and the three low bits, respectively. using a bitwise operator, W1 is embedded into the R byte by replacing its two LSB with the two bits of W1. Similarly, W2 is embedded into the G byte by replacing its three LSB with the three bits of W2, and W3 is embedded into the B byte by replacing its three LSB with the three bits of W3. The resulting embedded block is then converted back into pixels in RGB format and positioned in the same location as the original block from the host image, as shown in Figure 4.3.

4.2.1.2 Extraction watermark from medical image phase

The goal of the extraction phase is to recover the watermark from the watermarked image, decompose it into sub-watermarks, and reverse the rotation applied during

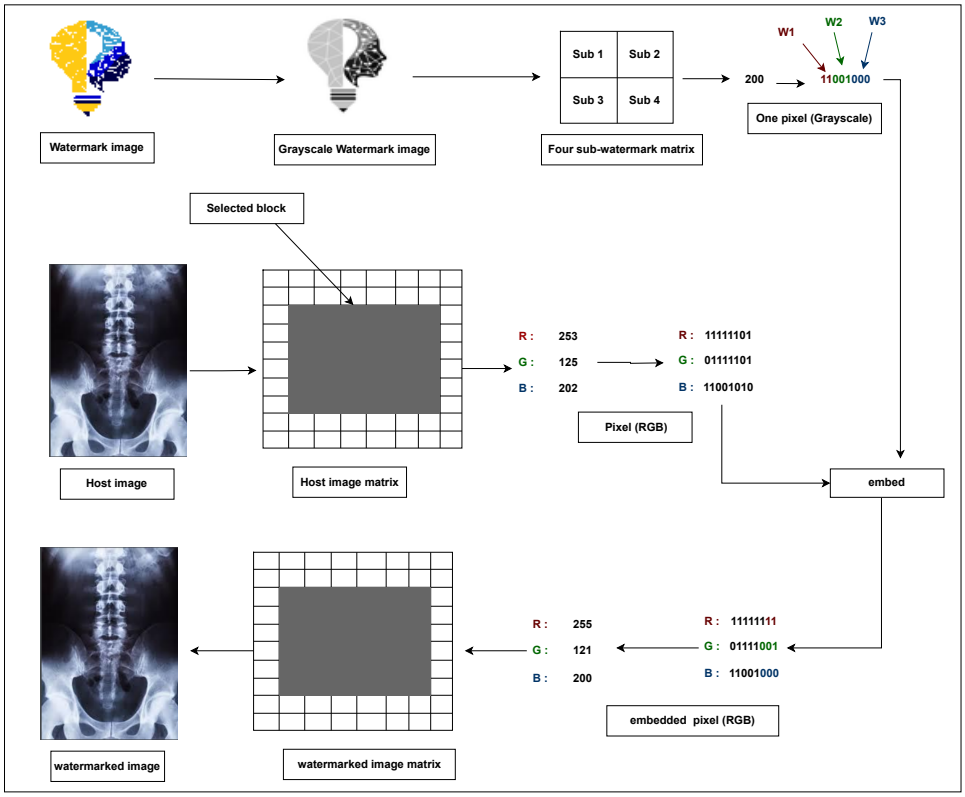


Figure 4.3: Embedding watermark image process.

embedding. This process involves four steps, as illustrated in Figure 4.4.

A block of pixels is extracted from the center of the image, matching the watermark's size. For each pixel in this extracted block, represented in RGB format (3 bytes), the watermark bits W1, W2, and W3 are extracted and combined to reconstruct a byte. These combined bytes are then organized into a matrix to recreate the grayscale watermark image. Finally, the grayscale watermark image is divided into four sub-watermarks.

4.2.2 Encryption layer

We utilized an asymmetric image encryption approach based on twin message fusion, as detailed in Chapter 3, which is derived from the El-Gamal algorithm [98], to encrypt the watermarked image. This method offers an effective homomorphic encryption technique through twin key encryption and magic number fragmentation. While MNF-G was initially developed for text encryption, this technique is now adapted for encrypting images.

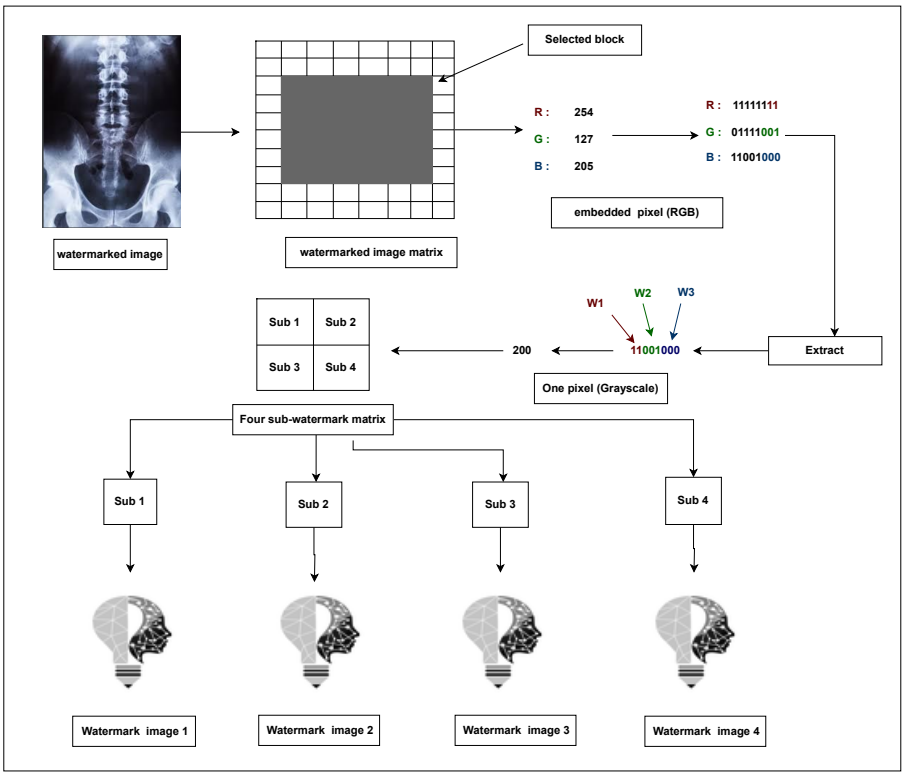


Figure 4.4: Extraction of sub-watermarks from the watermarked image process.

4.2.2.1 Watermarked medical image encryption and decryption processes

In the encryption process, the watermarked medical image is represented as a sequence of bits. The encryption procedure is illustrated in Figure 3.1 and is detailed in Algorithm 1. The same figure also depicts the decryption process, which is outlined in Algorithm 2.

Algorithm 1 Image Encryption Algorithm

Require: Watermarked image in binary format
Ensure: Encrypted image in RGB format
1: Treat the watermarked image as a sequence of bytes
2: Split the sequence of bytes into blocks of 32-bits
3: **for** each 32-bit block **do**
4: Encrypt the block using TMF to obtain an integer
5: Convert the resulting integer to binary format and append it to the encrypted byte sequence
6: **end for**
7: Reconstruct the encrypted image in RGB format from the encrypted byte sequence

Algorithm 2 Image Decryption Algorithm

Require: Encrypted image in binary format
Ensure: the Original watermarked image is in RGB format
1: Treat the encrypted image as a sequence of bytes
2: Split the sequence of bytes into blocks of 32-bits
3: **for** each 32-bit block **do**
4: Decrypt the block using TMF to obtain an integer
5: Convert the resulting integer to binary format and append it to the decrypted byte sequence
6: **end for**
7: Reconstruct the original watermarked image in RGB format from the decrypted byte sequence

4.3 Implementation

This section demonstrates a real-world experiment of our crypto-system within the IoMT architecture, as depicted in Figure 4.1. In this setup, the Raspberry Pi functions as the perception layer, acting as the sensor, while a laptop serves as the application layer, operating as the receiver.

The proposed crypto-system will be implemented and tested in two environments: IoT devices and a classical (laptop) environment, with the following specifications:

- **Raspberry Pi 3 Model B:**
  - CPU: Quad-Core 1.2GHz Broadcom BCM2837 64bit, ARM Cortex-A53 (ARMv8).
  - RAM 1GB.
  - Raspbian is the official operating system.
- **Laptop :**
  - CPU: Intel Core ™i3-8550u, CPU@1.80GHz Dual-core.
  - RAM 16GB.
  - Windows 10 operating system.

(a) (b) (c) (d) (e)

Figure 4.5: The result images of the crypto-system steps. (4.5a) The input image, (4.5b) Watermark, (4.5c) Watermarked image, (4.5d) Encrypted image, (4.5e) Decrypted image.

Perception Layer (Sender): The input image, referred to as the host or original image, is displayed in Figure 4.5a. For embedding the watermark, we used the image shown in Figure 4.5b. The medical image with the embedded watermark is illustrated in Figure 4.5c. To ensure secure transmission, the watermarked image is encrypted, resulting in the encrypted image shown in Figure 4.5d. These operations are performed on a Raspberry Pi device.

Network Layer: The connection between the Raspberry Pi and the laptop is established via a local WiFi network. This layer is essential for the secure transmission of encrypted images from the perception layer (sender) to the application layer (receiver).

Application Layer (Receiver): Upon receiving the encrypted image, the laptop decrypts it to retrieve the watermarked image, as shown in Figure 4.5e. Subsequently, the receiver extracts the watermarks to verify the authenticity and integrity of each sub-watermark, as depicted in Figure 4.6.

(a) (b) (c) (d)

Figure 4.6: Four extracted sub-watermarks from RW. (4.6a) Sub-watermark rotated 0°, (4.6b) Sub-watermark rotated -90°, (4.6c) Sub-watermark rotated -180°, (4.6d) Sub-watermark rotated -270°

4.4 Evaluation and Results

This section details the evaluation and results of the proposed crypto-system. It starts with a security analysis of the watermarking and encryption algorithms. An evaluation of the overall performance of the crypto-system in the IoMT environment is then provided. The section concludes with a comparative study.

4.4.1 Security validation

It is crucial to demonstrate that the proposed watermarking algorithm withstands various traditional attacks, such as Crop, Gaussian, Rotate, and Salt & Pepper. Table 4.1 provides details of these attacks, indicating that the watermark can be successfully extracted and identified under different conditions. Specifically, with a 33% crop, all four copies of the watermark can be extracted; at 50% crop, two copies are retrievable; and at 75% crop, one copy remains extractable. Despite some distortion caused by Gaussian noise and Salt & Pepper attacks, the watermarks remain significantly recognizable. Additionally, all four copies of the watermark can be extracted in all Rotate attacks. These results confirm that the watermarking algorithm is robust against traditional attacks.

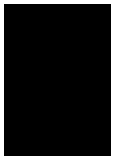
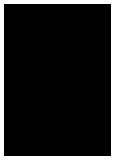
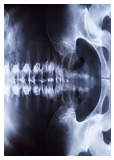
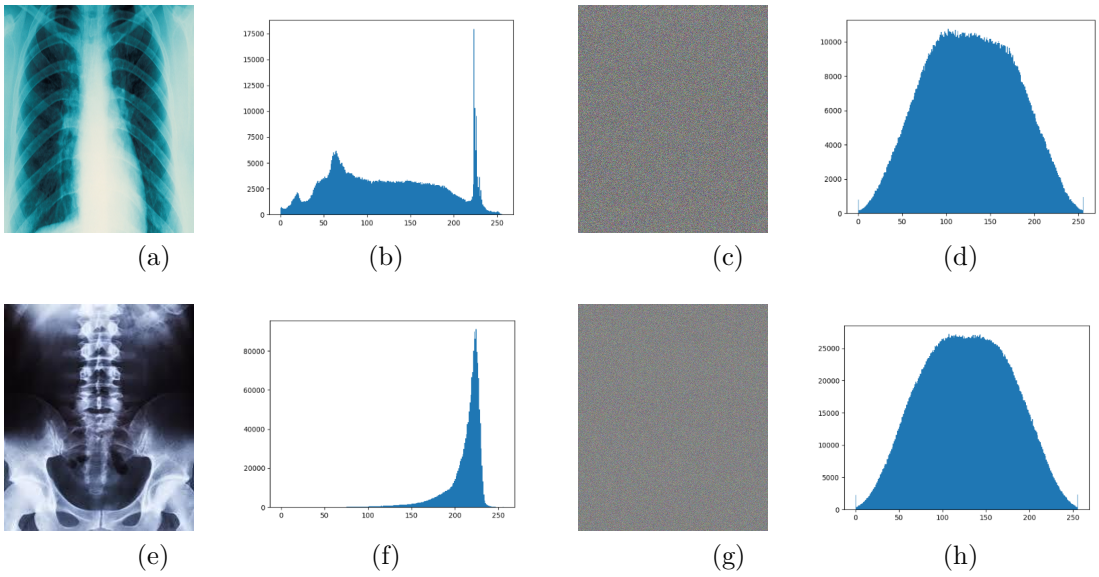
Attacks	Image Attacked	Sub-W 1	Sub-W 2	Sub-W 3	Sub-W 4
Crop 33 %					
Cut 75%					
Cut 50%					
Gaussian Noise					
Salt and Peppers					
Rotate 90					
Rotate 180					
Rotate 270					

Table 4.1: Sub-Watermark results for each kind of attack.

The histogram illustrates the distribution of pixel density in an image. The frequency distribution should be uniform for a well-encrypted image, without any distinct patterns or starting points, indicating that the data is thoroughly shuffled.

By comparing the histograms of the original images in Figures 4.7b and 4.7f with those of the encrypted images in Figures 4.7d and 4.7h, we can observe a clear and significant difference. This indicates that decrypting or tampering with the protected image is challenging.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.7: The Histograms of images, (4.7a),(4.7e) Origin images, (4.7b) , (4.7f) Histogram of origin images, (4.7c),(4.7g) Encrypted images, (4.7d) , (4.7h) Histogram of encrypted images.

4.4.1.1 Sybil and replay attack

An attacker can disrupt the network by reusing transmitted data to create redundant information, draining the sender's energy, or by impersonating the sender. To counteract this, we employ asymmetric encryption. Using a public key infrastructure (PKI), each user receives a certified public/private key pair. Only users with verified public keys can participate, ensuring that all interactions within the network are secure and authenticated.

4.4.1.2 Selective attack

The proposed scheme is susceptible to selective attacks, where an attacker might monitor the transceiver activity of IoT devices and selectively modify, block, or reorder messages to manipulate system behavior or deceive users. To bolster security, our crypto-system incorporates two protective layers: a watermarking layer and an encryption layer. The watermarking layer distributes four copies of the watermark throughout the entire host image, increasing the likelihood of retrieving the watermark even if parts are tampered with. Additionally, the watermarked image is encrypted, further obscuring the original data from attackers and making it more difficult for them to identify and alter the transmitted data.

4.4.2 IoMT performance evaluation of the proposed crypto-system

This subsection assesses the performance of our crypto-system in the IoMT environment, focusing on processing time, memory usage, and energy consumption. The evaluation was conducted using the aforementioned Raspberry Pi 3 Model B.

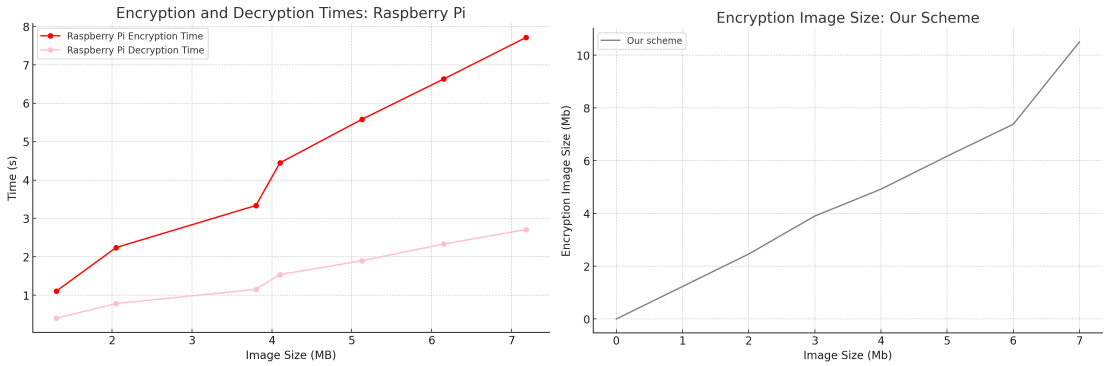
Processing time is a critical metric for evaluating the efficiency of a crypto-system. We measured the duration of each stage, including encryption and decryption. As shown in Figure 4.8a, the results demonstrate that our crypto-system operates efficiently within acceptable time frames for real-time applications in the IoMT environment, with minimal energy consumption.

Memory usage is another critical factor, particularly for resource-constrained IoMT devices. Our experiments demonstrated that the crypto-system uses memory efficiently without significantly straining the devices, as the increase in the size of the encrypted image compared to the original is minimal. Figure 4.8b illustrates this efficiency and scalability within the IoMT environment.

Overall, the performance evaluation shows that our crypto-system can effectively handle computational demands. Its ability to maintain low processing times and optimal memory usage makes it suitable for real-time applications in the IoMT domain.

4.4.3 Comparative study

In this section, we compare our crypto-system with several common techniques. Specifically, we evaluate the proposed watermarking technique against three recent methods: LSB [37], a DCT-based scheme [99], and a DWT-based scheme [100]. For



(a) (b)

Figure 4.8

Figure 4.9: The proposed crypto-system performance. (4.8a) Processing time of Crypto-system on IoMT environment (Raspberry Pi). (4.8b)Memory usage of Crypto-system on IoMT environment (Raspberry Pi).

the proposed encryption scheme, comparisons are made with both symmetric [52] and asymmetric [53] encryption schemes.

The comparison factors for the encryption scheme include the encryption and decryption execution times in two environments: IoT devices and a classical (laptop) environment. Additionally, entropy analysis is used to assess the encryption performance.

Attacks	**Proposed Work**	**LSB**	**DCT-based scheme**	**DWT-based scheme**
Gaussian Noise	34	33.51	36.65	31.01
Salt and Pepper	39.18	20.89	23.20	22.88
Cropping	55.12	15.24	18.11	23.14

Table 4.2: PSNR value of different algorithms for different attacks.

Entropy analysis is crucial in cryptography for ensuring the randomness and unpredictability of keys and data, thereby enhancing security and preventing attacks. The results of our entropy analysis are presented in Table 4.3. The proposed algorithm is compared with those in [53], [101], and [102]. For instance, when examining the encrypted X-ray images (1), (5), and (7), the corresponding entropies are 7.8288, 7.86239, and 7.841146, respectively. The higher entropy values of the proposed algorithm indicate greater security against entropy attacks compared to other methods.

	X-Ray Image 1	X-Ray Image 5	X-Ray Image 7
Proposed algorithm	7.82882	7.86239	7.841146
[53]	7.79595	7.80373	7.797613
[101]	7.4455	7.4446	7.5444
[102]	7.4982	7.5892	7.7563

Table 4.3: Entropy comparison of three images.

4.5 Conclusion

In this chapter, we developed an IoMT image crypto-system that integrates spatial watermarking with asymmetric encryption to secure medical image transfers within IoT's resource-constrained environment. The system ensures image authenticity and integrity through watermarking, followed by encryption to protect the image during transmission. Our watermarking algorithm demonstrated strong resilience against various attacks, outperforming other techniques in robustness and efficiency. Additionally, the encryption method provided a smaller encrypted image size and faster execution time compared to AES and RSA, making it well-suited for IoT applications.

Chapter 5

K-Means-Based Watermarking Algorithm For medical image

5.1 Introduction

Effective watermarking techniques are essential for ensuring the integrity and authenticity of medical images, which are critical assets in the healthcare industry [103]. In today's interconnected world, the rapid growth of digital information and the widespread sharing of medical images across various platforms have significantly increased the challenge of safeguarding these valuable assets [104]. As medical images are increasingly shared and accessed across networks, the risk of unauthorized modifications or misappropriation has become a major concern for researchers and healthcare professionals alike [105]. It has become imperative to develop advanced watermarking algorithms that can effectively protect these images from potential threats while maintaining their diagnostic quality.

To address these evolving challenges, this research proposes the development of an innovative watermarking algorithm specifically tailored for medical images. The new algorithm aims to overcome the shortcomings of existing techniques by providing a lightweight and highly efficient solution that guarantees the security and authenticity of medical images. At the same time, it minimizes any impact on the diagnostic quality of these images, ensuring that healthcare

5.2 Proposed technique

In this section, we delve into the proposed algorithm, which aims to protect and facilitate the exchange of medical images, ensuring data security and verifying sender authenticity with minimal impact on the image. This algorithm robustly detects and accurately identifies any alterations or tampering within the medical

image, thus ensuring the integrity and trustworthiness of the exchanged medical data.

The algorithm comprises four key steps: (1) generating a binary hash image from the watermark, (2) embedding the watermark into the host image, (3) extracting the watermark from a watermarked image, and (4) evaluating the watermark integrity by detecting any modifications introduced during the embedding process. By employing K-means clustering and utilizing Least Significant Bit (LSB) manipulation, this approach offers a reliable and secure method for protecting digital content with watermarks while allowing subsequent verification of data authenticity [106].

5.2.1 Generating a Hash Image

In the initial step, the algorithm converts the given watermark image into a two-dimensional array of pixels. It then uses K-means clustering to group similar pixel colors into eight clusters. Each cluster is replaced with its central point, reducing the image's complexity. The cluster center values are converted into binary format and concatenated to form a single binary string. This string is used to construct a binary hash image, which is replicated horizontally and vertically to match the dimensions of the host image. This hash image forms the basis for the embedding process in subsequent steps.

Algorithm 3 Generating a Hash Image

1: Convert the watermark image into a 2D array of pixels.
2: Apply K-means clustering to group pixel colors into 8 clusters.
3: Replace each cluster with its center point.
4: Convert cluster center values to binary format and concatenate them.
5: Create a binary hash image by repeating the binary string to match the host image size.

5.2.2 Embedding the Watermark

In the second step, the algorithm embeds the watermark into the host image. First, it ensures that the dimensions of the watermark hash image match those of the host image. The algorithm retrieves the LSB from the green channel for each pixel in the host image and replaces it with the corresponding bit from the watermark hash image. This process discreetly embeds the watermark within the host image, preserving data authenticity while minimizing visual impact or alteration to the original image.

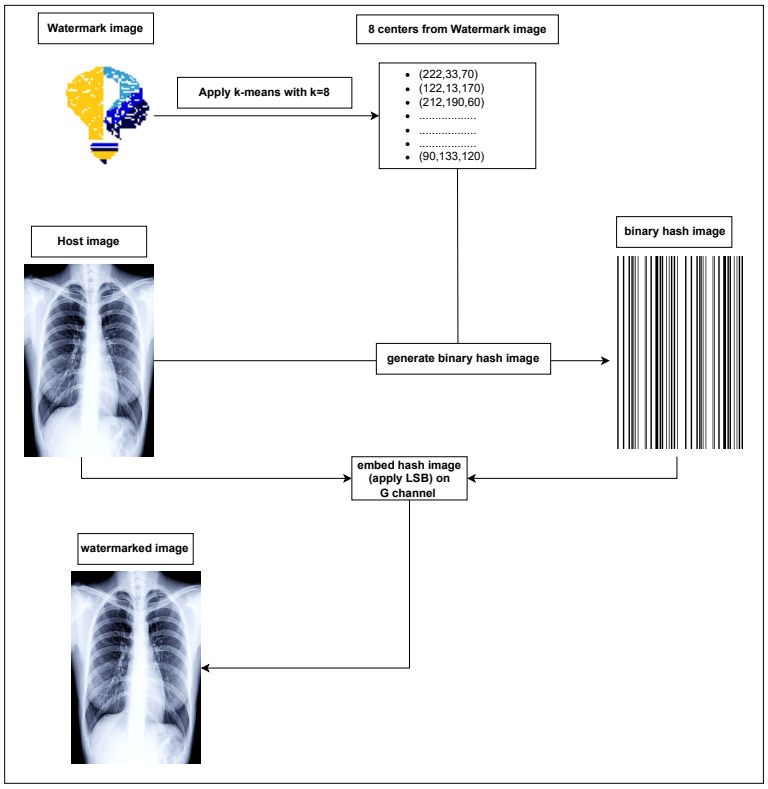


Figure 5.1: Illustration of Generating a Hash Image and Embedding the Watermark

5.2.3 Extracting the Watermark

In the third step, the algorithm retrieves the embedded watermark from the watermarked image. It initializes an empty array to store the extracted watermark. The algorithm examines each pixel within the watermarked image, extracting the LSB from the green channel and storing these bits in the watermark image array. This careful procedure ensures the accurate retrieval of the watermark while preserving the integrity of the original data.

Algorithm 4 Extracting the Watermark

1: Initialize an empty array to store the extracted watermark.
2: **for** each pixel in the watermarked image **do**
3: Retrieve the LSB of the green channel.
4: Store the extracted bit in the watermark image array.
5: **end for**

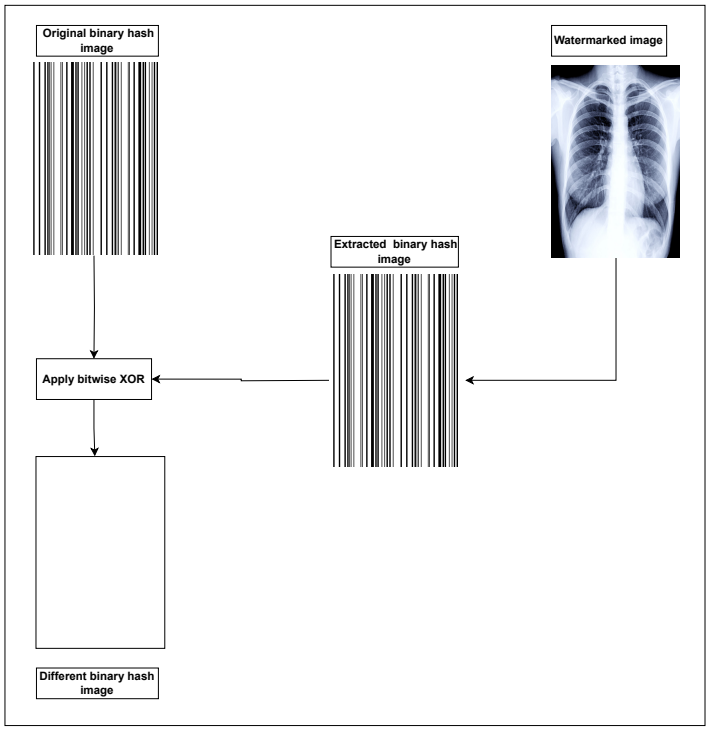


Figure 5.2: Illustration of Extracting the Watermark and Evaluating Watermark Integrity

5.2.4 Evaluating Watermark Integrity

In the final step, the algorithm assesses the robustness of the watermark and its resistance to tampering. This evaluation involves calculating the XOR (exclusive OR) difference between the original binary hash image generated in the first step and the extracted watermark from the third step. If the XOR result is 1, it indicates that a pixel was modified during the embedding process, suggesting potential tampering. Conversely, a result of 0 indicates that the pixel remains unaltered, affirming the data's integrity. This crucial step enables the algorithm to verify the authenticity and integrity of the watermarked image, ensuring it has not been tampered with during transmission or storage [107].

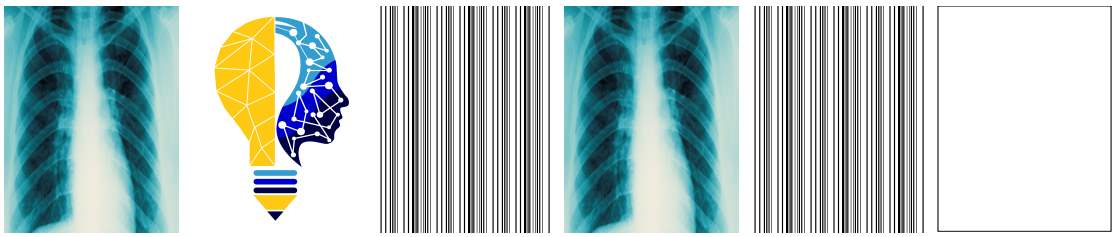
Algorithm 5 Evaluating Watermark Integrity

1: Calculate the XOR difference between the original binary hash image and the extracted watermark.
2: **if** XOR result is 1 **then**
3: Pixel was modified during embedding.
4: **else**
5: Pixel remains unaltered.
6: **end if**

5.3 Implementation

In this section, we detail the experimental setup and present the results of our proposed watermarking algorithm, utilizing X-ray medical images to show its effectiveness. The system was implemented in Python and executed on the Windows 10 operating system. with an Intel® Core™ i7 CPU, and 8GB of RAM.

For our experiments, we utilized the NIH Chest X-rays medical image dataset [108]. This extensive dataset includes a wide range of chest radiographs annotated for various thoracic abnormalities. preprocessing, which involved image resizing and standardization, was conducted to prepare the dataset for watermarking.



(a) (b) (c) (d) (e) (f)

Figure 5.3: The image results of the watermarking algorithm steps. (5.3a) The host image, (5.3b) The watermark image, (5.3c) The binary hash image, (5.3d) The watermarked image, (5.3e) The extracted binary hash image, (5.3f) The XOR result image

Figure 5.3 shows the various result images from the steps of the watermarking algorithm. Initially, we use two input images: the 'host image' (Figure 5.3a) and the 'watermark image' (Figure 5.3b). A binary hash image is generated for the embedding process, as shown in Figure 5.3c. After embedding the binary hash image, the resultant watermarked medical image is shown in Figure 5.3d.

The extracted binary hash image from the watermarked image is shown in Figure 5.3e. In the final step, the bitwise XOR operated the extracted and previously generated binary hash images. Figure 5.3f illustrates the resulting binary hash image, where any tampered pixels are indicated by black pixels.

5.4 Evaluation and Results

The evaluation of our proposed watermarking technology for medical images involved subjecting the scheme to various types of attacks. Additionally, we compared our scheme with related methods using Peak Signal-to-Noise Ratio values [109].

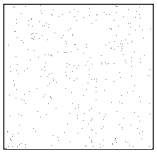
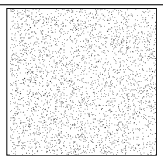
Attacks	Watermarked image	XOR binary hash image
Text Attack		
Salt and Peppers (0.05)		
Salt and Peppers (0.1)		
Cut 25%		

Table 5.1: The image results of our scheme under different attacks

5.4.1 Attacks Analysis

Table 5.1 shows the result of subjecting our scheme to various types of attacks and noise to assess its effectiveness in detecting tampering and identifying affected areas.

In the first attack, which introduced textual content into the images, our scheme demonstrated exceptional accuracy in detecting and localizing the added text. This capability illustrates the robustness of our algorithm in identifying alterations caused by such attacks. The next attack involved adding salt and pepper noise at different levels. Our scheme effectively identified tampered pixels, with its performance becoming particularly evident when the noise rate was increased to 0.1. At this higher noise level, tampering was visible, further highlighting the scheme's ability to detect significant disturbances in the image accurately. In the final attack, our scheme was evaluated for its response to image cropping, specifically removing 25% of the image. Our algorithm identified the cropped portion, demonstrating its sensitivity to such manipulation and precision in locating the alterations.

These results collectively highlight our scheme's effectiveness in detecting and pinpointing tampering, underscoring its robustness against various forms of image manipulation.

Method	PSNR (dB)
Our method	54.69
Scheme in [110]	44.51
Scheme in [111]	51.53
Scheme in [63]	49.62
Scheme in [112]	47.01

Table 5.2: Comparative PSNR Values for Image Watermarking Methods

5.4.2 Comparative study

Based on Table 5.2, our scheme proves to be the most effective approach for watermarking images compared to various alternative schemes as demonstrated by PSNR values. Our scheme achieved an impressive PSNR of 54.69 dB, significantly outperforming the other schemes. Specifically, the Scheme in [110] scored 44.51 dB, the Scheme in [111] reached 51.53 dB, the Scheme in [63] achieved 49.62 dB, and the Scheme in [112] obtained 47.01 dB.

The technique's PSNR and that of the other techniques highlight its higher ability to maintain image quality during watermarking. Therefore, our technique appears as a promising choice for watermarking technique, offering a higher level of fidelity and robustness, and making it the preferred option for applications requiring high-quality watermarked images.

The result demonstrates the efficacy of our proposed watermarking technology for medical images, accurately detecting and locating tampering without causing noticeable changes. This capability is essential in medical imaging, aiding in precise and reliable diagnoses. In conclusion, watermarking technology is crucial for ensuring the integrity and authenticity of medical images.

5.5 Conclusion

In this thesis, we introduced a K-Means-based watermarking algorithm designed to secure medical images by ensuring their authenticity and integrity. The algorithm uses an XOR binary hash image to detect any pixel manipulation, demonstrating strong resilience against various attacks, including Text, Salt and Pepper, and Cutting. The PSNR analysis confirmed that the watermarking process has a minimal impact on the host image. Comparative studies further highlighted the algorithm's superiority and its effectiveness in detecting manipulations, making it a robust solution for securing medical images.

Chapter 6

Blockchain-Watermarking scheme based K-Means for Medical Image

6.1 Introduction

Effective watermarking techniques are crucial for securing medical images, which are fundamental to healthcare delivery. As the digitization of medical images becomes more prevalent in diagnosis, treatment planning, and patient care, the need to safeguard these digital assets has become increasingly urgent [113],. The extensive sharing and storage of medical images across various platforms have raised significant concerns about unauthorized access, alteration, and misuse [114]. Ensuring the authenticity and protection of these images is essential, as any tampering could lead to serious consequences, such as misdiagnosis and compromised patient safety.

Blockchain technology offers a promising solution to these challenges with its decentralized, traceable, and immutable nature [42]. By ensuring that each node in the network maintains a complete copy of the data ledger, blockchain significantly reduces the risks of data loss, system failure, or hacking. This research proposes an innovative approach that integrates blockchain technology with K-means clustering to enhance the security and efficiency of medical image watermarking. This method not only eliminates the need for third-party verification but also ensures that the diagnostic quality of the images is preserved, addressing the specific security needs of the healthcare industry.

6.2 Proposed technique

This section provides a comprehensive explanation of the proposed scheme, which aims to secure and facilitate the exchange of medical and watermark images without relying on a third party. The scheme is designed to ensure data integrity and

verify the sender's authenticity, all while minimizing the impact on the image.

The proposed scheme consists of two phases. The first phase is divided into four steps, as illustrated in Figure 6.1: (1) generating an actual image from the watermark, (2) embedding the actual watermark into the host image, (3) uploading the actual watermark to the IPFS network, and (4) storing the hash value obtained from the IPFS network on the blockchain using a smart contract.

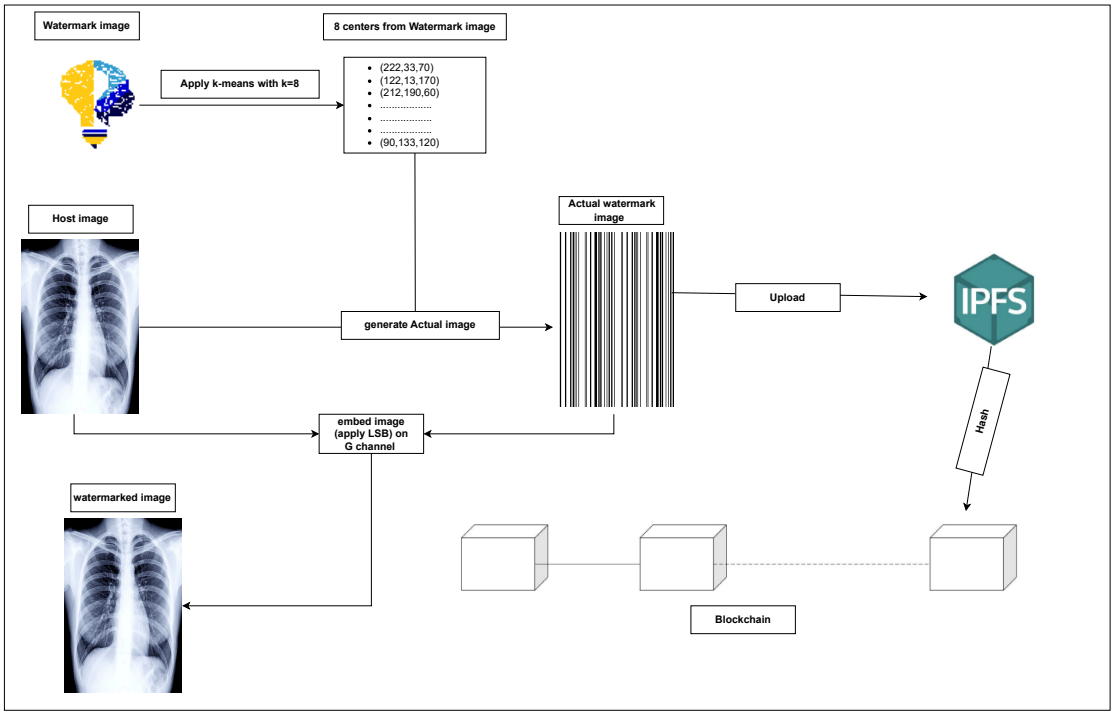


Figure 6.1: Steps of the First Phase in Our Scheme

6.2.1 Phase 1: Generating, embedding, and uploading the watermark

The algorithm starts by converting the provided watermark image into a 2D-pixel array. It then applies K-means clustering to group similar pixel colors into eight clusters. Each pixel is replaced by the centroid of its corresponding cluster. The cluster centers' values are converted to binary format and concatenated to form a single binary string. This string is used to generate an actual image, which is then duplicated horizontally and vertically to match the size of the target host image as shown in Algorithm 6.

In the second step, the LSB is used to embed the actual watermark. The LSBs from the green channel of each pixel in the host image are extracted and replaced with

bits from the watermark image. This process embeds the watermark discreetly, preserving the host image's integrity and appearance.

Algorithm 6 Watermark Actual Image Generation

1: Transform the watermark image into a two-dimensional pixel array.
2: Utilize K-means clustering to categorize pixels into eight clusters based on color.
3: Substitute the pixels in each cluster with the centroid of that cluster.
4: Convert the values of the cluster centroids into a binary format and merge them into a single binary string.
5: Construct an actual image by replicating the binary string horizontally and vertically to fit the dimensions of the target host image.

In the final steps, the actual watermark image is uploaded to the IPFS network, which returns a string of hash values. These hash values are then uploaded to the blockchain using a smart contract, ensuring the information remains protected in its encrypted state. As shown in Figure 6.1, this scheme guarantees the long-term safety and authenticity of the protected content.

6.2.2 Phase 2: Downloading, Extracting, and Verification of the Watermark

The second phase is dedicated to the retrieval and verification of the watermark, comprising four steps: (1) retrieving the hash value from the blockchain, (2) downloading the actual watermark from the IPFS network, (3) extracting the watermark from the watermarked image, and (4) verifying the integrity of the watermarked image by comparing the downloaded actual watermark with the extracted watermark. These steps are illustrated in Figure 6.1.

A smart contract uploads a file to the blockchain, generating a hash value that serves as the transaction's contract address. To extract the watermark from the blockchain, this contract address must be entered to locate the associated contract. Using the hash value stored within the contract, the uploaded actual watermark can then be downloaded from the IPFS network.

In the third step, the algorithm retrieves the embedded watermark from the watermarked image. It starts by examining each pixel, specifically extracting the LSB from the green channel of each pixel. These bits are then carefully collected and arranged in an array to reconstruct the watermark image.

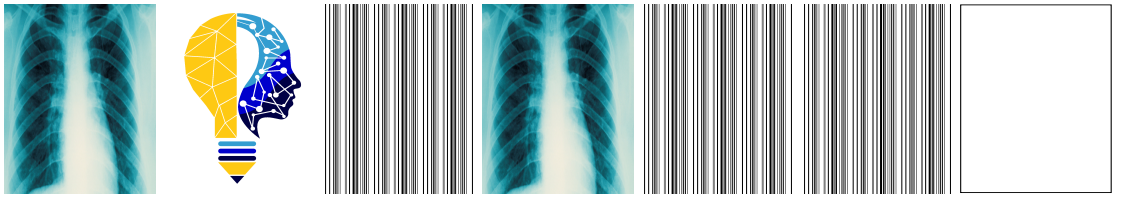
In the final step, the algorithm evaluates the robustness of the watermark and its strength to modifications. This involves applying the XOR (exclusive OR)

operation between the downloaded actual watermark and the extracted watermark. If the result of this operation is 1, it indicates that a pixel position in the watermarked image has been altered, suggesting potential tampering. If the result is 0, it means the pixel remains unchanged, verifying the integrity of that pixel position. This important process allows the scheme to accurately determine the authenticity and integrity of the watermarked image, ensuring its protection against alterations during transmission or storage.

6.3 Implementation

This section details the experiments of the proposed scheme. For the experimental analysis, we used the NIH Chest dataset [108] s. This comprehensive dataset includes chest radiographs, each annotated to identify various thoracic abnormalities.

The scheme was implemented in Python and deployed on a PC with an Intel® Core™ i7 processor, 8GB of RAM, a 240GB SSD, and Windows 10 as the operating system.



(a) (b) (c) (d) (e) (f) (g)

Figure 6.2: The image results of the watermarking algorithm steps. (6.2a) The host image, (6.2b) The watermark image, (6.2c) The binary hash image, (6.2d) The watermarked image, (6.2e) The extracted binary hash image, (6.2f) The downloaded actual watermark binary image, (6.2g) The XOR result image

Figure 6.2 shows the various result images at different stages of our scheme. The host image used in this experiment is depicted in Figure 6.2a, while the watermark image is shown in Figure 6.2b. Applying Algorithm 6 to both the watermark image and the host image generates the actual watermark, as illustrated in Figure 6.2c.

The generated actual watermark is uploaded to the IPFS network in the following steps, which returns a unique hash string. This hash value is then stored on the blockchain through a smart contract. The outcomes of these steps are summarized in Table 6.1. The upload process generates information including the file name, the hash value from the IPFS network, the transaction hash, the address of the

Steps	Details
Uploading an actual watermark to IPFS	Name: 'watermak-image.png'; Hash: 'QmcUWGszPf9aX5X7CxgUriM6dWZo5Smg6q.....'; Size: '1204'
Uploading the hash value to the blockchain by writing a smart contract	Status: 0x1 Transaction mined and execution succeed; Transaction Hash: 0x2f4e1bb3e8f6c2e..c4b2158a6b26c....; From: 0x4E17DaF6a7....3ECbA75e56fefD34; To: secure.docs(string) 0x8e5bCCe52FEDf256e917e482E....; Gas: 3200000 gas; Transaction Cost: 90500 gas; Execution Cost: 67220 gas; Hash: 0x2f4e1bb3e8f6c.....c4b6d88e4a6b65; Input: 0xa5b...00000; Decoded Input: ["string documentHash": "QmcUWGszPf9aX5X7CxgU.."]; Decoded Output: ; Logs: []; Value: 0 wei

Table 6.1: Uploading Actual Watermark and Blockchain Transaction By Writing a Smart Contract

uploading account, and details on execution and transaction costs. The watermark stored on the blockchain is securely protected and immutable.

In the final phase, after receiving the watermarked image from the sender, the watermark binary image is extracted using the extraction algorithm. The resulting extracted image is shown in Figure 6.2e. The next step involves retrieving the IPFS hash value from the blockchain through a smart contract and then downloading the original actual watermark from the IPFS network. This downloaded watermark image is illustrated in Figure 6.2f.

6.4 Evaluation and Results

This section evaluates the validity and accuracy of our proposed Blockchain-Watermarking scheme for medical images. We applied various attacks and different types of noise to assess the scheme's effectiveness in identifying tampering and locating the manipulation regions.

In the initial attack, as illustrated 6.2, the noise was added as textual content. Our method detected and accurately located the inserted text, as demonstrated

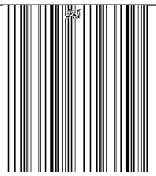
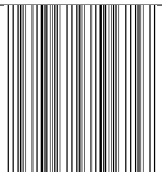
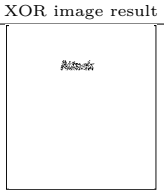
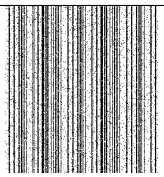
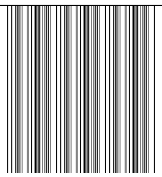
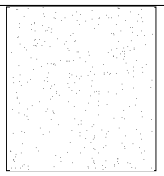
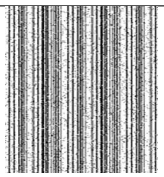
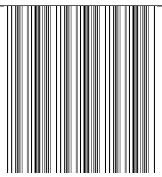
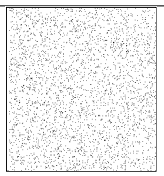
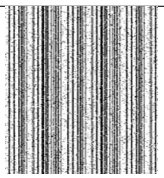
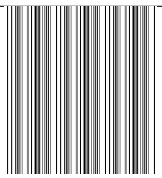
Attacks	Watermarked image	Extract watermark	Actual watermark	XOR image result
Text Attack				
Salt and Peppers (0.05)				
Salt and Peppers (0.1)				
Cut 25%				

Table 6.2: The image results of our scheme under different attacks

in the XOR image result. In subsequent attacks, such as the introduction of salt and pepper noise at different levels, our approach showed strong robustness, accurately identifying the locations of the altered pixels. When the noise level in the watermarked image reached 0.1, the tampered areas became more visible in the XOR image. Furthermore, our method effectively identified the cropped area in the final attack, which affected 25 percent of the image. These results highlight the sensitivity and precision of our method in detecting and locating altered or tampered regions within images.

6.4.1 Comparative study

This subsection provides a comparative analysis of our scheme against other related works, focusing on PSNR values. The impact of watermark embedding on image quality was evaluated by comparing the PSNR of our watermarked image with those generated by several well-known methods [115, 63, 110, 111, 112]. As illustrated in Table 6.3, our scheme demonstrated superior image quality, achieving an impressive PSNR of 54.69 decibels. This PSNR is notably higher than the values achieved by other methods, with [110] reporting 44.51 dB, [111] recording 51.53 dB, [63] obtaining 49.62 dB, [112] registering 47.01 dB, and [115] reaching 52.89 dB.

Method	PSNR (dB)
Our Proposed Scheme	54.69
Scheme in [115]	52.89
Scheme in [63]	49.62
Scheme in [110]	44.51
Scheme in [111]	51.53
Scheme in [112]	47.01

Table 6.3: Comparative PSNR Values for Image Watermarking Methods

6.5 Conclusion

In this thesis, we introduced a Blockchain-Watermarking scheme based on K-Means, designed to secure medical image transfers without relying on third parties. The scheme ensures image authenticity and integrity by embedding a generated watermark into the host image, with its hash stored securely on the blockchain via IPFS. Our evaluation demonstrated the algorithm's resilience against attacks like Text, Salt and Pepper, and Cutting, with PSNR analysis confirming minimal impact on the host image. This approach effectively enhances the security and integrity of medical image exchanges.

Chapter 7

A Nature-Inspired Partial Distance-Based Clustering Algorithm

7.1 Introduction

Effective clustering techniques are crucial for extracting valuable insights from large and complex datasets, which is increasingly important in today's data-driven world [116]. Traditional clustering methods like hierarchical and partitional techniques have been foundational in organizing data [76, 117], uncovering patterns, and supporting decision-making. However, as data volume and dimensionality continue to grow, these methods face significant challenges in maintaining computational efficiency and accuracy, particularly in high-dimensional spaces and large-scale datasets.

To overcome these challenges, this research introduces a novel clustering algorithm called K-level, inspired by the collective behavior of fish locomotion. The K-level algorithm efficiently merges clusters based on distances, creating hierarchical structures without exhaustive iterations, thus reducing computational complexity. This innovative approach enhances flexibility and scalability across various data types, making it an effective solution for accurate and efficient clustering in increasingly complex data environments.

7.2 Proposed technique

The K-level algorithm is inspired by the collective movement patterns observed in fish. It translates this natural behavior into a computational clustering method. Just as fish navigate their surroundings through coordinated group movements,

the K-level algorithm iteratively merges clusters based on their closeness, reflecting the synchronized movements of fish in a collection.

At each step, the K-level algorithm identifies and merges clusters closest to each other, much like how fish group together in response to external factors or common traits. This distance-based merging process allows K-level to build hierarchical structures efficiently, avoiding the need for exhaustive calculations, similar to the streamlined coordination seen in a collection of fish. K-level provides a scalable and flexible clustering method suitable for various datasets and applications. This section will define and describe the K-level algorithm in detail, thoroughly explaining its mechanics and functionality.

7.2.1 Definitions

Definition 1 (Dataset)**.** *A dataset is a collection of data points, represented by*

$$C = \{x_0, x_1, \ldots, x_n\},$$

where C denotes the dataset, x_i represents the ith data object within the dataset, and n indicates the total number of data objects in C. Each data object is characterized by a set of features with a dimensionality of d, defined as:

$$x_i = [f_i^1, f_i^2, \ldots, f_i^d].$$

Definition 2 (Cluster)**.** *A cluster is a group of data points that are close to each other based on a defined set of features. It is denoted by c. The closeness of data points within a cluster is typically determined by a similarity measure, such as the Euclidean similarity distance.*

Definition 3 (Cluster Center)**.** *A Cluster Center refers to the central point of a cluster. For clusters that are being merged, the cluster center x is determined through a recursive merging process, which is represented as follows:*

$$x = ((((x_1 \oplus x_2) + x_3) + \ldots) + x_m) \tag{7.1}$$

where x_i represents the centers of the clusters to be merged, and m denotes the number of clusters. The operator $\oplus$ is defined as follows:

$$x_i \oplus x_j = (f_i^1 + f_j^1, f_i^2 + f_j^2, \ldots, f_i^d + f_j^d).$$

Definition 4 (Clustering Levels)**.** *Clustering Levels refer to the distinct stages within the clustering process, which divide the process into sequential steps, allowing for the exploration of inherent correlations within the data. These levels are denoted by k, where the maximum level is determined by the dataset size and the*

number of clusters merged. At each stage, the number of clusters to be merged is represented by m.

Definition 5 (Euclidean Similarity Distance)**.** *Euclidean similarity distance is a metric used to quantify the similarity or dissimilarity between two data points in a multi-dimensional space. It is defined as:*

$$\textit{distance}(x_i, x_j) = \sqrt{(f_i^1 - f_j^1)^2 + (f_i^2 - f_j^2)^2 + \cdots + (f_i^d - f_j^d)^2} \quad \textit{where } x_i, x_j \in C \tag{7.2}$$

Here, f_i^1, f_i^2, …, f_i^d are the coordinates of point x_i, and f_j^1, f_j^2, …, f_j^d are the coordinates of point x_j, with d representing the dimensionality of the data points.

K-Level Operation

This section details the procedural steps of the K-level clustering algorithm, which operates across multiple stages (levels). The algorithm organizes a dataset into a hierarchical structure of clusters spanning k levels, using similarity measures as the basis for clustering.

At level 0, the algorithm begins with the dataset $C = x_0, x_1, \ldots, x_n$, where each data point is treated as an individual cluster. For illustrative purposes, to visualize the process, the dataset is represented by a set of 2D points, with each $x_i = (f_i^1, f_i^2)$, as depicted in Figure 7.1.

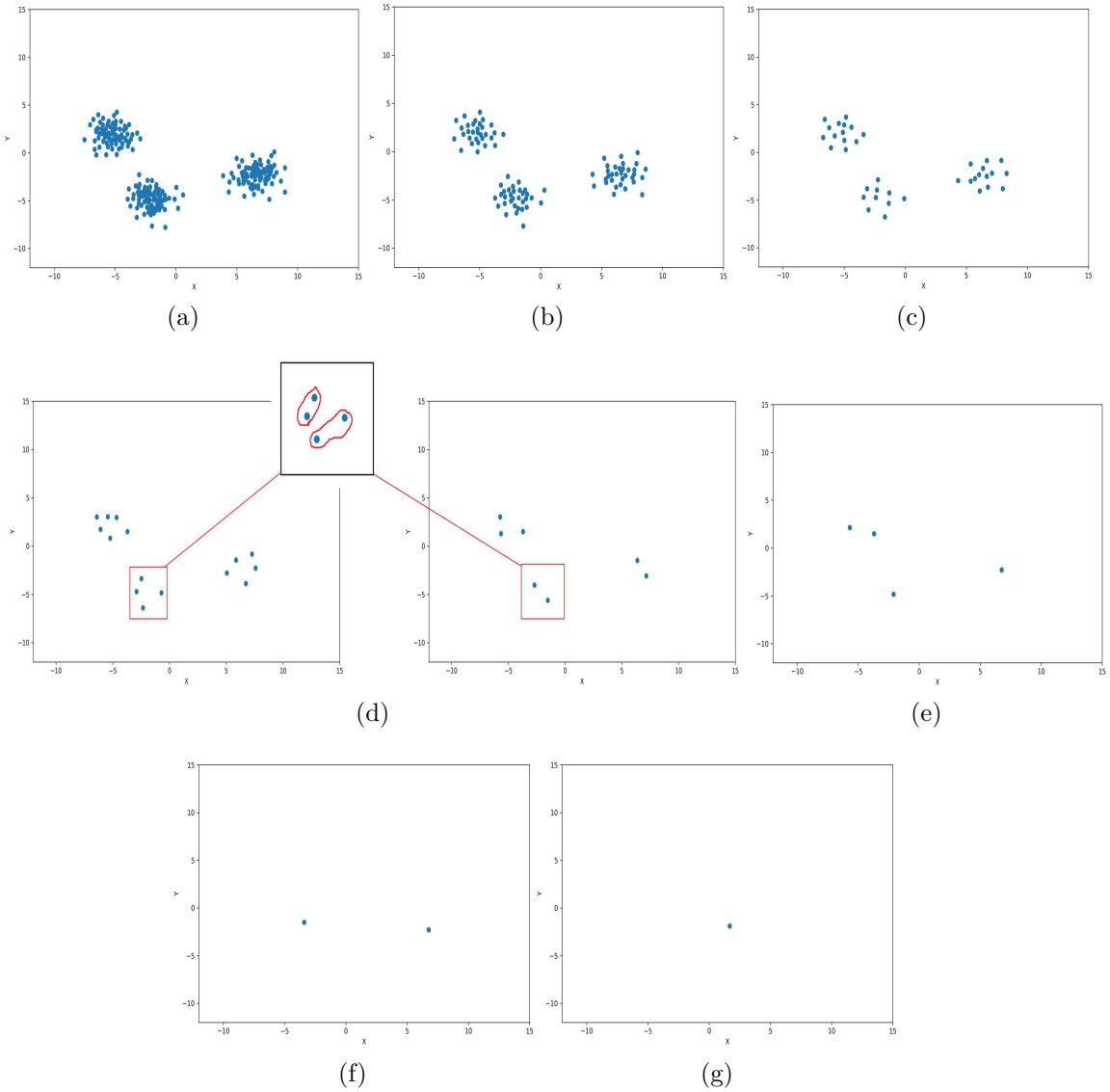


Figure 7.1: Graphical representation of K-level (k = 7 and m = 2). (**a**) Level 0: Initialization; (**b**) level 1;(**c**) level 2; (**d**) level 3 and level 4; (**e**) level 5; (**f**) level 6; (**g**) level 7.

In level 1 of the algorithm, the data point x_0 is selected, and a distance array $D(x_i)$ is constructed for x_0 by calculating the Euclidean distances between x_0 and all other items in the set C. The resulting distance array $D(x_i)$, which has a size of $n-1$, is defined as follow

$$D(x_i) = \{d(x_i, x_j) | \forall x_j \in C, x_j \neq x_i\} \tag{7.3}$$

The term $d(x_i, x_j)$ represents the Euclidean similarity distance between the cluster

points x_j and x_i. Beginning with the point x_0, this Euclidean similarity distance measures the distance between x_0 and each cluster x_i within the set C_0, excluding x_0 itself.

After constructing the distance array $D(x)$ for x_0, the algorithm selects the m clusters with the smallest distances from x_0. These selected clusters (points) are then grouped into a new cluster. The new cluster's center calculating using the equation specified in Equation (7.1).. Once merged, these clusters are removed from the initial dataset C_0. The centers of the newly formed clusters are added to the dataset for the next level, C_1. This process continues iteratively until C_0 is empty. The algorithm progresses from level 0 to level k (from C_0 to C_k), systematically building a hierarchical structure of clusters across multiple levels.

As the K-level algorithm advances to higher levels ($k > 0$), the clusters formed in the previous level are treated as individual clusters, and the same merging process is repeated. This iterative procedure continues until the algorithm reaches the desired level k. The steps of the proposed K-level algorithm are outlined in **Algorithm 7** as follows:

Algorithm 7 : The K-level Algorithm

Input:

- $C = \{x_0, x_1, ..., x_n\}$ Dataset.
- k desired level.
- m number of clusters to merge.

Output: Hierarchical clustering structure with k levels
Begin

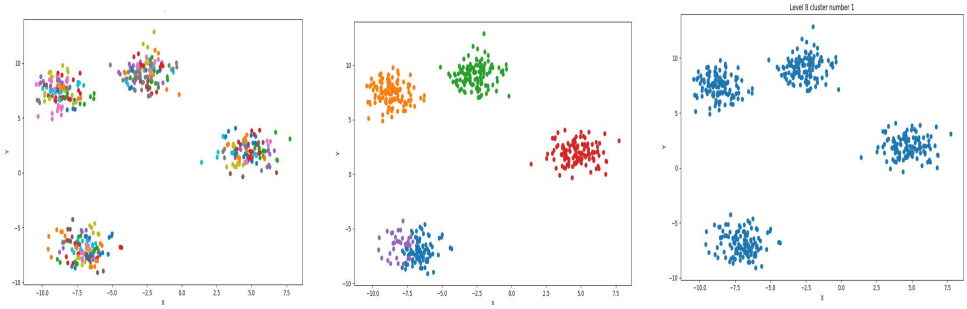
1. Initialize C_0 where each data point is considered an individual cluster from dataset C.
2. For i in range from 1 to k:
   (a) Initialize an empty cluster set C_i
   (b) while C_{i-1} is not empty:
      i. select random x_j from C_{i-1}.
      ii. Construct the distance array $D(x_j)$ for x_j with other clusters in C_i using Equation (7.3).
      iii. Select m clusters with the smallest distances based on $D(x_j)$.
      iv. Merge the m selected clusters by calculating their center using Equation (7.1).
      v. Add the merged cluster to C_i.
      vi. Remove the m merged clusters from C_i.
3. Return the hierarchical clustering structure with k levels: $C_1, C_2, ..., C_k$

End

7.3 Implementation

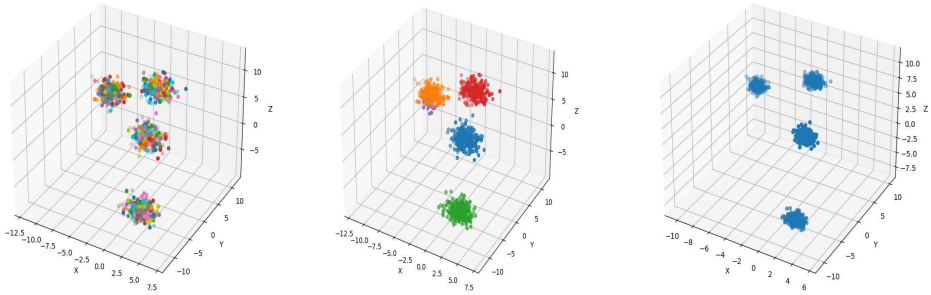
This section presents the experiments and results of the proposed K-level. Variant synthetic datasets are used in this work to demonstrate the effectiveness of K-level.

The proposed algorithm was implemented in Python (the code is available on GitHub [118]) and executed on a personal computer equipped with an Intel® Core™ i5-8550U CPU @ 1.80 GHz and 8 GB of RAM, running Windows 10. To assess the performance of the proposed method, the clustering algorithm was applied to various synthetic datasets. Figures 7.2–7.5 illustrate the clustering results across different levels, dataset sizes, and numbers of clusters.



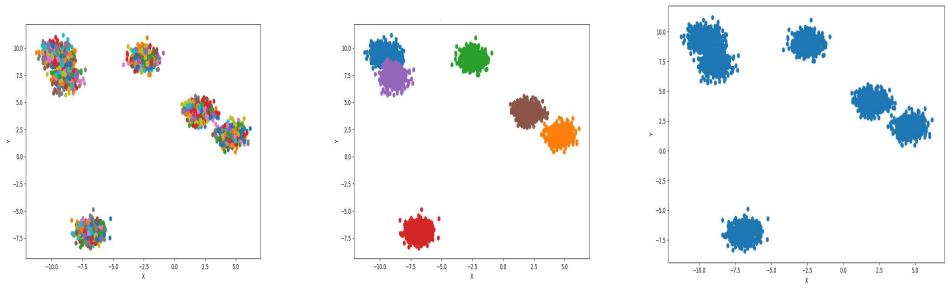
(a) (b) (c)

Figure 7.2: Result of clustering dataset (6000) in 2 dimensions with 4 clusters (Each cluster has a different color). (**a**) Level 1; (**b**) level 4; (**c**) level 6.



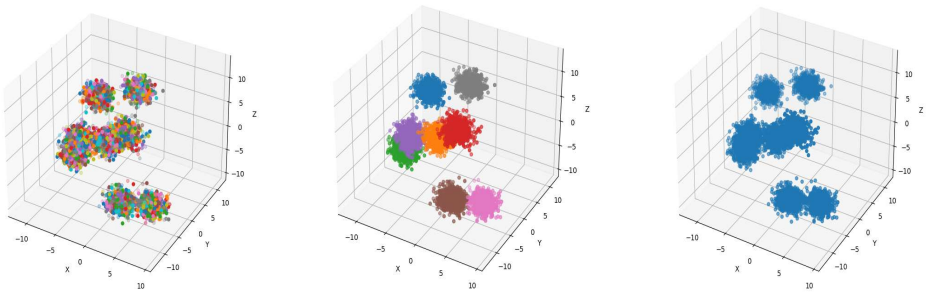
(a) (b) (c)

Figure 7.3: Result of clustering dataset (1000) in 3 dimensions with 4 clusters (Each cluster has a different color). (**a**) Level 1; (**b**) level 4; (**c**) level 6.



(a) (b) (c)

Figure 7.4: Result of clustering dataset (40,000) in 2 dimensions with 6 clusters (Each cluster has a different color). (**a**) Level 1; (**b**) level 4; (**c**) level 6.



(a) (b) (c)

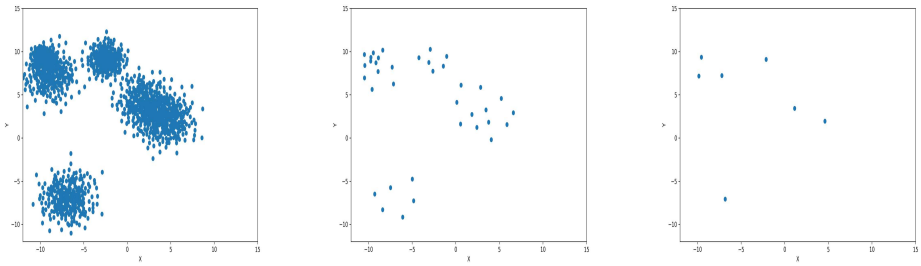
Figure 7.5: Result of clustering dataset (6000) in 3 dimensions with 8 clusters (Each cluster has a different color). (**a**) Level 1; (**b**) level 4; (**c**) level 6.

Figures 7.2a, 7.3a, 7.4a, and 7.5a illustrate level 1 of the clustering process, where each cluster is initially merged with its nearest neighbor, resulting in a reduction in the total number of clusters. As the algorithm progresses to level 2, the number of clusters decreases further as additional nearest-neighbor clusters are merged. By level 4, shown in Figures 7.2b, 7.3b, 7.4b, and 7.5b, the number of clusters align with the original number of groups from which the dataset was formed. This iterative process continues at each level until all clusters are eventually merged into a single cluster, as depicted in Figures 7.2c, 7.3c, 7.4c, and 7.5c. These results demonstrate the algorithm's correctness and accuracy in handling datasets with varying structures, including differences in the number of dimensions and the number of groups within the dataset.

7.4 Evaluation and Results

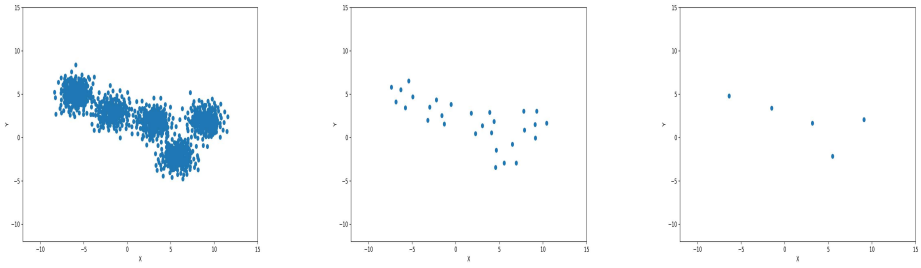
This section evaluates the effectiveness and accuracy of our algorithm, beginning with its ability to reduce the size of variant datasets. We then compare the performance of our algorithm variants with those of related works.

To reduce the dataset size for machine learning and other applications, the K-level algorithm uses cluster centers to represent all elements within each cluster, regardless of the cluster's size. This approach enables the algorithm to represent the dataset without altering its overall meaning or the distribution of elements within it. To demonstrate this concept and its outcomes, the algorithm was applied to six synthetic datasets with distinct data distributions, as shown in Figures 7.6–7.11.



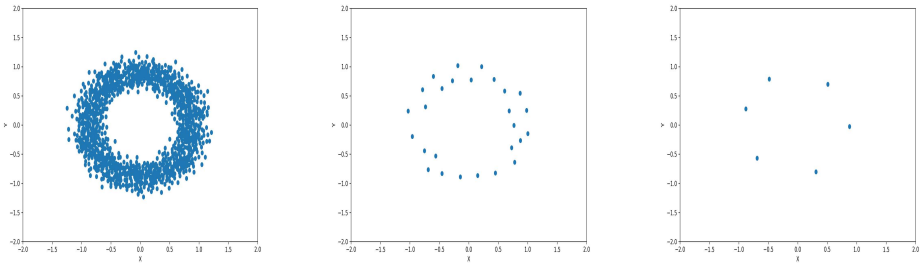
(a) (b) (c)

Figure 7.6: Results of Center Points (Anisotropic Dataset, 4000 Points). (**a**) Level 1; (**b**) level 4; (**c**) level 6.



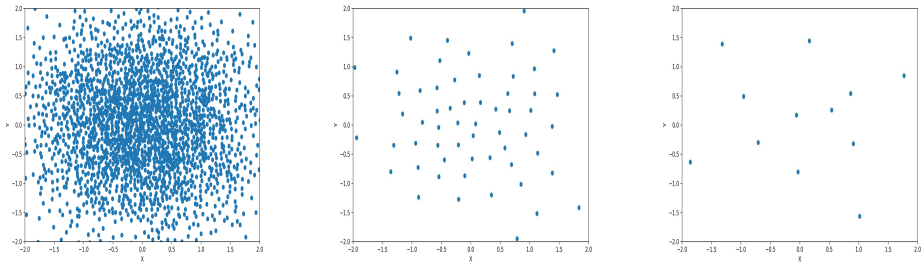
(a) (b) (c)

Figure 7.7: Results of Center Points (Blob Dataset, 3000 Points). (**a**) Level 1; (**b**) level 4; (**c**) level 6.



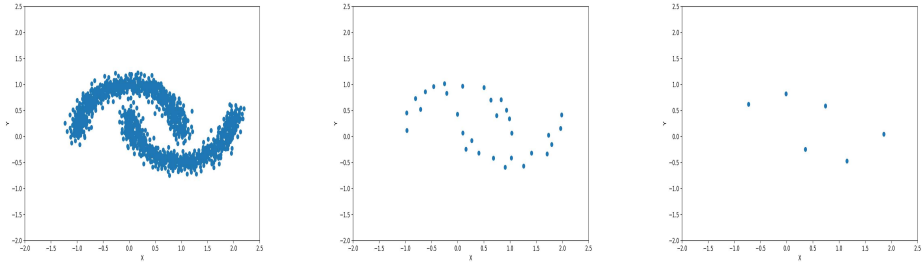
(a) (b) (c)

Figure 7.8: Results of Center Points (Circles Dataset, 3000 Points). (**a**) Level 1; (**b**) level 4; (**c**) level 6.



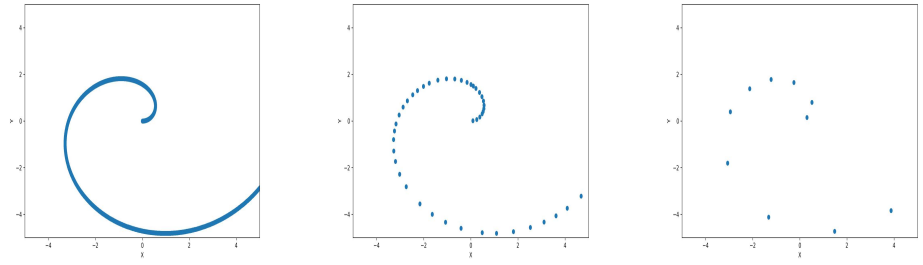
(a) (b) (c)

Figure 7.9: Results of Center Points (Gaussian Dataset, 6000 Points). (**a**) Level 1; (**b**) level 4; (**c**) level 6.



(a) (b) (c)

Figure 7.10: Results of Center Points (Moon Dataset, 3000 Points). (**a**) Level 1; (**b**) level 4; (**c**) level 6.



(a) (b) (c)

Figure 7.11: Results of Center Points (Spiral Dataset, 3000 Points). (**a**) Level 1; (**b**) level 4; (**c**) level 6.

The results depicted in the figures demonstrate that as the level of consolidation increases, the number of data points decreases while preserving the overall shape and distribution of the elements within the dataset. By level 6, as shown in Figures 7.6c, 7.7c, 7.8c, 7.9c, 7.10c, and 7.11c, the dataset size is significantly reduced, yet the general structure of the distribution remains recognizable. This indicates that

the reduced dataset, generated at different levels, can effectively represent the original dataset. These reduced data points can be used in various applications, producing results that closely resemble those obtained from the complete dataset.

7.4.1 Comparative study

In this section, we assess the effectiveness and accuracy of the K-level algorithm through a comparative analysis with several established clustering methods, including UPGMA [75], CLINK [71], UPGMC [74], SLINK [70], and K-means [119]. We employed internal validation metrics such as the Silhouette Score [120], Davies–Bouldin Index [121], and Calinski–Harabasz Index [122] to evaluate the quality and performance of these clustering algorithms. These metrics provide valuable insights, enabling a comprehensive comparison and thorough evaluation of the proposed K-level algorithm across various datasets, as demonstrated in Tables 7.1–7.4.

Table 7.1: Comparison of clustering algorithms on Blob dataset (the bold characters are the best values).

Method	**Blob**		
	Silhouette Score	**Davies–Bouldin Index**	**Calinski–Harabasz Index**
K-level	0.61	0.50	**12,868.1**
UPGMA	**0.64**	**0.45**	11,889.9
CLINK	**0.64**	**0.45**	12,093.73
UPGMC	0.55	0.45	12,032.22
SLINK	0.36	0.67	2456.5
K-means	0.61	0.50	12,512.4

Table 7.2: Comparison of clustering algorithms on Circles dataset (the bold characters are the best values).

Method	**Circles**		
	Silhouette Score	**Davies–Bouldin Index**	**Calinski–Harabasz Index**
K-level	0.38	**0.66**	**4305.6**
UPGMA	0.41	0.68	3223.7
CLINK	0.38	0.69	2671.0
UPGMC	**0.43**	0.67	3399.3
SLINK	−0.22	0.68	2.15
K-means	0.45	0.65	3853.2

Table 7.3: Comparison of clustering algorithms on Moon dataset (the bold characters are the best values).

Method	Moon		
	Silhouette Score	**Davies–Bouldin Index**	**Calinski–Harabasz Index**
K-level	**0.44**	**0.72**	**4807.3**
UPGMA	0.39	0.84	3509.3
CLINK	0.38	0.84	3562.9
UPGMC	0.42	0.79	3894.3
SLINK	−0.38	1.89	1.23
k-means	0.44	0.79	4490.9

Table 7.4: Comparison of clustering algorithms on Spiral dataset (the bold characters are the best values).

Method	Spiral		
	Silhouette Score	**Davies–Bouldin Index**	**Calinski–Harabasz Index**
K-level	0.52	**0.48**	8593.2
UPGMA	0.54	0.58	5064.9
CLINK	0.54	0.58	5064.9
UPGMC	0.54	0.58	5064.9
SLINK	0.31	0.51	3.3
k-means	**0.56**	0.61	**8871.9**

The K-level clustering algorithm demonstrates strong performance relative to other clustering methods across various datasets. In the following discussion, we compare the results of the K-level algorithm with those of other clustering approaches.

In the Blob dataset, the K-level algorithm achieves a strong Silhouette score of approximately 0.61, indicating well-defined clusters. The Davies–Bouldin Index, at 0.50, further confirms effective cluster separation and pattern recognition. Notably, K-level outperforms UPGMA and CLINK in the Calinski–Harabasz index, scoring 12,868.1 compared to their 11,889.9. This higher Calinski–Harabasz index emphasizes K-level's ability to form distinct clusters, showcasing its effectiveness as a clustering method for this dataset. Consequently, K-level is particularly well-suited for datasets with compact, well-separated clusters, making it an optimal choice for identifying distinct, non-overlapping groups.

In the Circles dataset, the K-level algorithm shows a lower Silhouette score of 0.38 and a Davies–Bouldin index of 0.66, suggesting relatively weaker cluster separation. However, it still performs reasonably well with a Calinski–Harabasz index of 4305.6, indicating some level of cluster differentiation. Although K-level has a

slightly lower Silhouette score compared to UPGMA and CLINK (0.38 vs. 0.41), it compensates by outperforming them in the Calinski–Harabasz index (4305.6 vs. 3223.7). This performance suggests that K-level has the potential to uncover patterns in the Circles dataset. Therefore, K-level is well-suited for datasets with less distinct clusters, making it a viable option for situations where clusters may overlap or have a more complex structure.

In the Moon dataset, the K-level algorithm achieves a notable Silhouette score of 0.44, reflecting well-defined clusters. It also registers a Davies–Bouldin Index of 0.72, indicating strong cluster separation and effective pattern recognition. Particularly impressive is K-level's performance in the Calinski–Harabasz index, where it scores 4807.3, significantly surpassing UPGMA and CLINK, which scored 5064.9. This superior result in the Calinski–Harabasz index underscores K-level's ability to form well-defined clusters, positioning it as a robust clustering method for the Moon dataset. In conclusion, K-level is well-suited for scenarios where the dataset contains well-separated clusters with clear boundaries, making it highly effective for tasks requiring precise cluster identification.

In the Spiral dataset, the K-level algorithm shows moderate clustering quality, achieving a Silhouette score of 0.52. It also excels in cluster separation, as indicated by a Davies–Bouldin Index of 0.48. The high Calinski–Harabasz index of 8593.2 further signifies well-defined clusters. While K-level's Silhouette score is close to that of UPGMA and CLINK (0.52 vs. 0.54), it outperforms them significantly in the Calinski–Harabasz index (8593.2 vs. 5064.9), highlighting its ability to generate more distinct clusters in the Spiral dataset. This establishes K-level as an effective clustering method for this type of data. Therefore, K-level is well-suited for datasets where clusters may not be perfectly compact but still exhibit discernible patterns, making it a strong choice for problems involving clusters with varying shapes and sizes.

The observed results across various datasets highlight the K-level algorithm's ability to produce well-defined clusters, thanks to its distinct features such as precise distance measurement, fitness criteria, greedy merging, hierarchical structure formation, and adaptive clustering levels. These characteristics enable the algorithm to effectively identify and merge the closest clusters, as evidenced by the high Silhouette scores and Calinski–Harabasz index values in the Blob and Moon datasets. The fitness criteria, which consider intra-cluster similarity and inter-cluster separation, ensure robust performance, particularly demonstrated by the Davies–Bouldin index values that indicate good separation in the Blob and Spiral datasets. The algorithm's greedy merging process enhances computational efficiency and scalability, which is reflected in the moderate-to-high Calinski–Harabasz index values across different datasets. The hierarchical structure formation is particularly beneficial for managing complex datasets with varying shapes, as seen in the Moon and Spiral datasets. Furthermore, the option for users to specify the number of

clusters to merge at each level provides greater control and adaptability, making the K-level algorithm an effective clustering method even in more challenging scenarios, such as those presented by the Circles dataset.

In terms of complexity, as shown in Table 7.5, the K-level algorithm demonstrates lower time complexity compared to traditional hierarchical clustering methods like UPGMA, CLINK, UPGMC, SLINK, and Median-link. K-level's time complexity is expressed as $O(n^3/m^3)$, whereas the other algorithms generally have a time complexity of $O(n^3)$. Notably, as the value of m increases, the time complexity of K-level decreases, enhancing its efficiency.

Table 7.5: Comparison of the time and memory complexities for clustering algorithms n is the size of the dataset, m number of clusters to merge simultaneously).

Method	**Time Complexity**	**Memory Complexity**
K-level	$O(n^3/m^3)$	$O(n)$
UPGMA	$O(n^3)$	$O(n^2)$
CLINK	$O(n^3)$	$O(n^2)$
UPGMC	$O(n^3)$	$O(n^2)$
SLINK	$O(n^3)$	$O(n^2)$
Median-link	$O(n^3)$	$O(n^2)$

K-level demonstrates a favorable memory complexity, denoted as $O(n)$, which is significantly lower than that of other clustering algorithms listed in Table 7.5. This efficient memory utilization makes K-level an attractive option, particularly in scenarios where minimizing memory usage is critical and when working with large datasets.

We analyze the total computation time across different dataset sizes for various clustering algorithms, including UPGMA, CLINK, UPGMC, SLINK, and K-means, as illustrated in Figure 7.12.

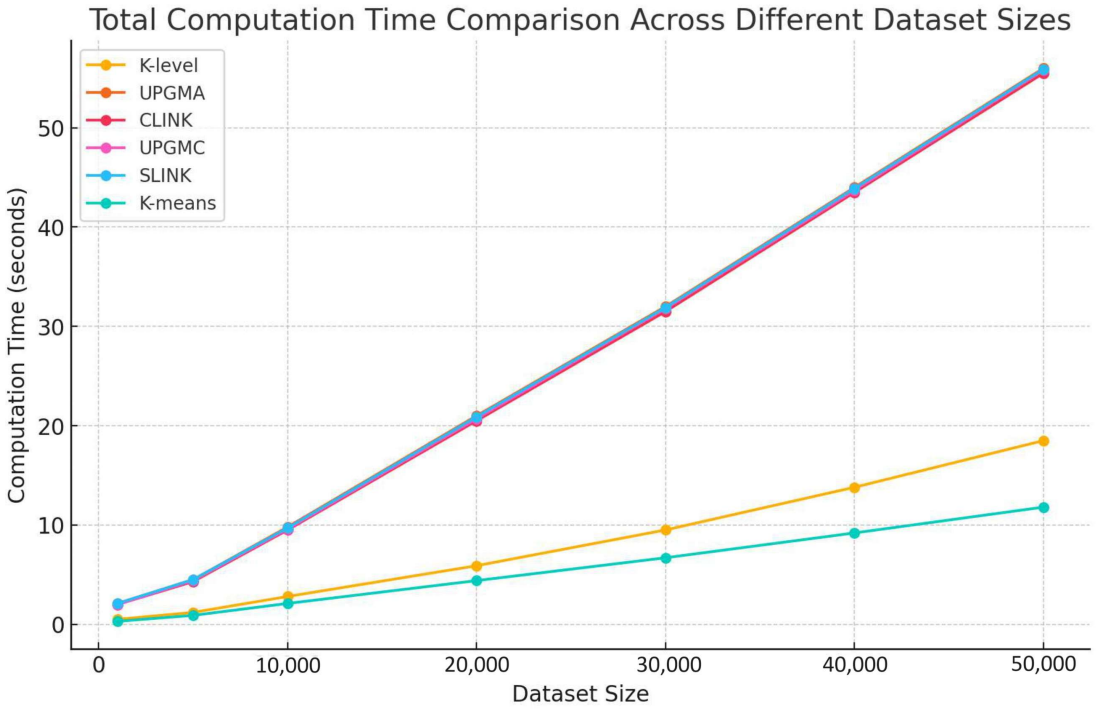


Figure 7.12: Comparison of the total computation time across different dataset sizes.

Our analysis reveals notable differences in computation time among the clustering algorithms as dataset sizes grow. The K-level algorithm consistently shows the shortest computation time across all dataset sizes. In contrast, traditional hierarchical algorithms such as UPGMA, CLINK, UPGMC, and SLINK exhibit significant increases in computation time as dataset sizes increase, highlighting their higher computational complexities. The similarity in computation time between the K-level algorithm and K-means suggests that K-level experiences only a moderate rise in computation time, making it more efficient than traditional hierarchical methods.

The K-level algorithm's superior performance in terms of computation time across various dataset sizes underscores its lower computational demands and better scalability compared to traditional hierarchical methods. This performance validates the design and efficiency of the K-level algorithm, establishing it as a robust and efficient clustering solution for managing large-scale datasets. These qualities make the K-level algorithm particularly well-suited for modern, data-intensive applications, highlighting its potential for widespread adoption in real-world scenarios where processing efficiency and scalability are critical.

7.5 Conclusion

In this thesis, we introduced a nature-inspired K-level clustering algorithm that overcomes the limitations of traditional hierarchical clustering methods. The K-level algorithm efficiently manages computational and memory resources by merging clusters in stages based on partial distances, demonstrating its ability to handle diverse data types and generate new data with similar patterns. Comparative analysis showed its superiority over other clustering algorithms in terms of internal validation metrics, with lower time and space complexity. These results validate the K-level algorithm as a robust solution for clustering problems with varying shapes and sizes. Future research will focus on applying the algorithm to real-world datasets and exploring its potential in dynamic and online clustering scenarios.

Chapter 8

A Novel Dual Watermarking Scheme Based On K-level For Medical Images

8.1 Introduction

Effective watermarking techniques are essential for safeguarding the security and integrity of medical images, which are critical to reliable healthcare delivery. As digital data continues to proliferate, the widespread sharing and storage of medical images introduce significant challenges in protecting these valuable assets [12, 8]. The increasing volume of data heightens concerns about maintaining the authenticity and confidentiality of medical images. Advanced watermarking algorithms have become more necessary than ever to address these issues. However, despite the availability of existing solutions, the risk of unauthorized access, tampering, and misuse remains a serious concern, potentially leading to severe consequences such as misdiagnosis.

Traditional watermarking methods, including frequency domain techniques like DWT and DCT, offer robust protection by embedding watermarks into the data's frequency coefficients [15]. However, these methods tend to be complex and slow. Conversely, spatial domain techniques, such as LSB insertion, provide simplicity and speed but lack the resilience to withstand sophisticated attacks [123]. As digital threats evolve, conventional methods face increasing difficulty in balancing security with maintaining image quality. To overcome these challenges, our research proposes a novel dual watermarking scheme that integrates multiple watermarking techniques with K-level clustering. This approach aims to enhance both the security and efficiency of medical image protection while preserving diagnostic quality, addressing the growing demands for advanced healthcare data security solutions.

8.2 Proposed technique

This section describes the proposed technique to protect and facilitate the exchange of medical images. This technique ensures data integrity and authenticates the sender with minimal impact on the image quality.

The method consists of three phases. In the first phase, there are three steps: (1) creating a binary image from the watermark, (2) embedding the original watermark into the host image using Discrete Cosine Transform (DCT), and (3) embedding the binary image into the watermarked host image. The second phase involves two steps: (1) extracting the binary image, and (2) retrieving the original image. In the final phase, the integrity of the data is verified using the binary image.

8.2.1 Phase 1: Embedding Dual Watermarking

The proposed watermark embedding scheme begins by converting the given watermark image into a two-dimensional (2D) pixel array. The pixel array undergoes K-level clustering, which groups similar pixel colors into 32 distinct clusters. Each pixel in a cluster is then replaced by the centroid of that cluster. The values of these cluster centers are converted into binary format. These binary values are concatenated to form a unified binary string, which is then used to generate a binary image. This binary image is subsequently expanded both horizontally and vertically to match the dimensions of the target host image.

Next, DCT is applied to the host image. The watermark image is embedded into this transformed host image. Then, the IDCT is used to convert the host image back to its spatial domain, now with the embedded watermark.

In the final step, the binary image is embedded into the previously watermarked image using the LSB technique, resulting in a dual watermarked image. This phase is illustrated in Figure 8.1.

8.2.2 Phase 2: Extracting Dual Watermarking

In this phase, the scheme retrieves the embedded watermarks from the watermarked image using a two-step process.

The first step involves extracting the binary watermark. This is achieved by examining each pixel in the watermarked image, specifically focusing on the green

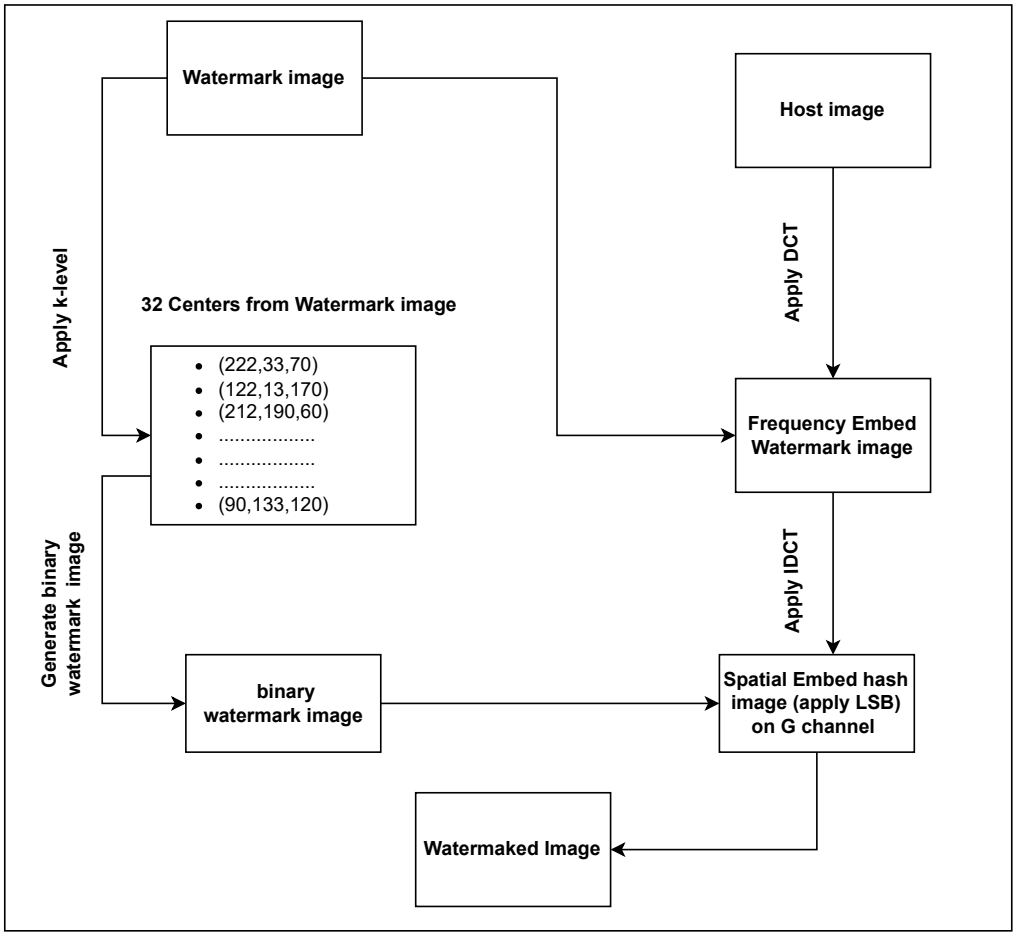


Figure 8.1: Phase 1: Embedding dual Watermarking

channel. The LSB from the green channel is extracted from each pixel and compiled into an array, which is then used to reconstruct the binary watermark image.

In the second step, the scheme applies the DCT to the watermarked image. This converts the image from the spatial domain to the frequency domain, making it possible to identify the watermark embedded within the DCT coefficients. The extraction process isolates the watermark based on these coefficients, effectively retrieving the second watermark. This process is illustrated in Fig 8.2

8.2.3 Phase 3: verification watermark

In the final phase, the algorithm verifies the robustness of the embedded watermark against modifications. This is achieved by performing an XOR operation between

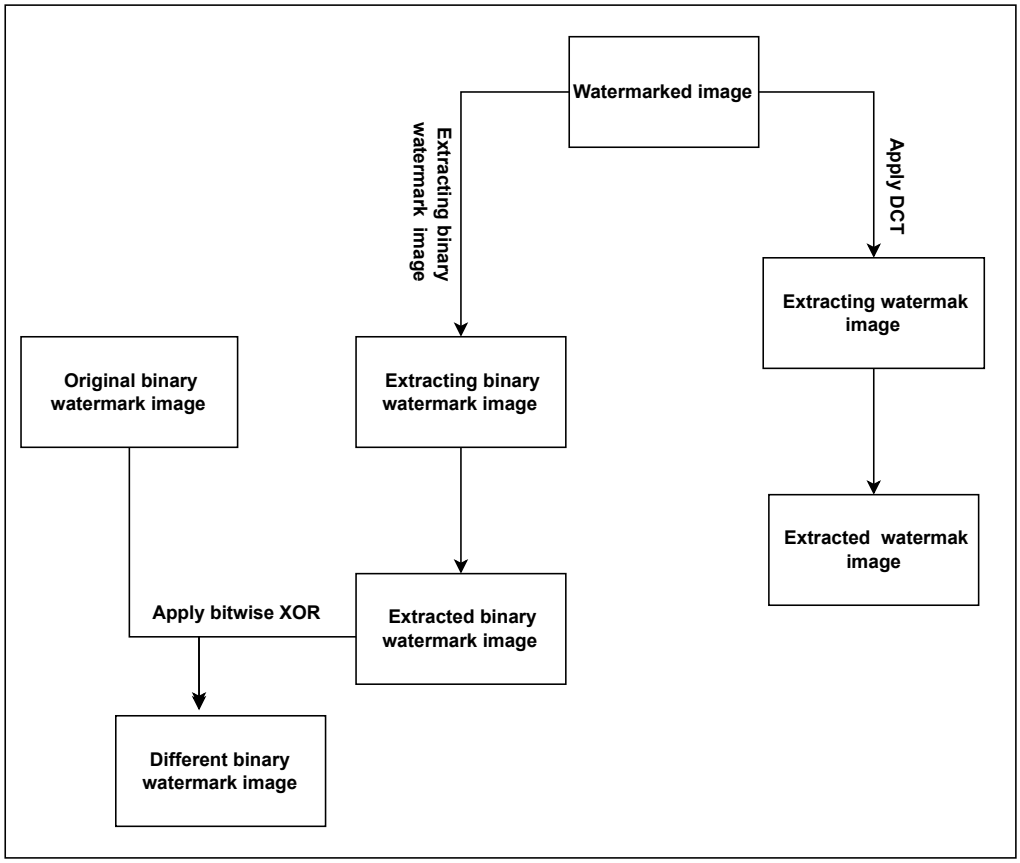


Figure 8.2: Phase 1: Extracting dual watermarking

the original binary watermark (embedded during the watermarking process) and the extracted watermark (recovered from the watermarked image). This verification step is essential for determining if the watermarked image has been altered during transmission or storage. By comparing the original and extracted watermarks through the XOR operation, the algorithm can confirm the authenticity and integrity of the watermarked image, enhancing its security against unauthorized changes.

8.3 Implementation

In this section, we describe the experiment of our proposed watermarking algorithm, which utilizes X-ray medical images to demonstrate its effectiveness. The

system was implemented in Python and ruined on a Windows 10 operating system, for our experiments, we utilized also the NIH Chest X-rays medical image dataset [124].

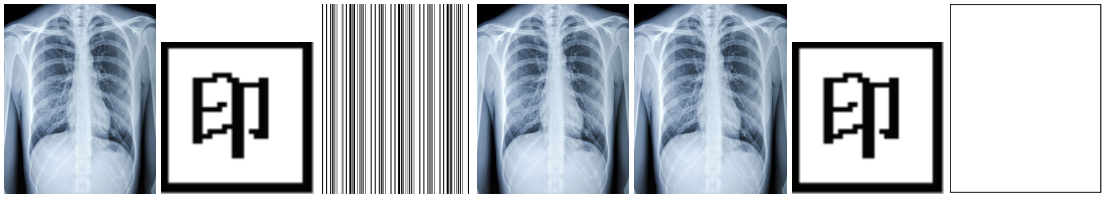


Figure 8.3: Experimental results of the watermark algorithm steps. (a) The host image;(b) The watermark image; (c) The binary watermark image; (d) The watermarked image; (e) The extracted binary watermark image;(f) The extracted watermark image; and (g) The XOR result image.

Figure 8.3 presents the various images result at each phase of our watermarking technique. The host image for this experiment is depicted in Figure 8.3a, while the watermark image is shown in Figure 8.3b. The binary image was generated from the watermark image according to the dimensions of the host image, as illustrated in Figure 8.3c in the first step. The watermark embedding step starts with embedding the watermark Figure 8.3b, followed by a second embedding step where the binary watermark Figure 8.3c is embedded into the image obtained from the first embedding step. The resulting dual watermarked image is displayed in Figure 8.3d.

The watermark extraction phase involves several steps. The binary watermark is extracted from the host image, as shown in Figure 8.3e, and the second watermark is obtained by applying DCT to the watermarked image, resulting in the image displayed in Figure 8.3f.

The final phase involves tamper detection, which is achieved by performing an XOR operation between the original binary watermark and the extracted binary watermark. This XOR operation reveals any differences between the two, facilitating tampering detection, as illustrated in Figure 8.3g. If there are no discrepancies, the XOR result will be a blank image (all zeros), indicating that no tampering has occurred; any deviations will suggest unauthorized modifications to the watermarked image.

8.4 Evaluation and Results

To evaluate our proposed dual watermarking scheme for medical images, we perform various types of attacks on this scheme. Furthermore, we compared our scheme with related methods by using PSNR values.

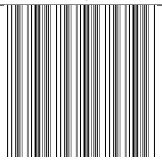
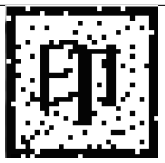
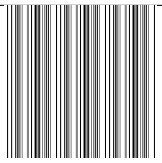
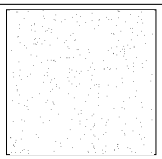
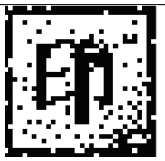
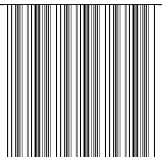
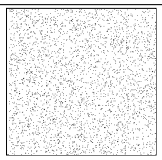
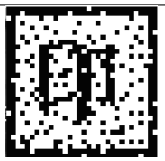
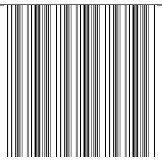
Attacks	Watermarked image	Extract watermark	Extract binary watermark	XOR image result
Text Attack				
Salt and Peppers (0.05)				
Salt and Peppers (0.1)				
Cut 25%				

Table 8.1: The image results of our scheme under different attacks

Our scheme effectively detected and located the added text in the first attack, which involved embedding noise as textual content. This is evidenced by the XOR image result and the extracted watermark, as shown in Table 8.1. Similar results were obtained when salt and pepper noise was introduced at varying rates; our scheme proved robust by accurately identifying the positions of tampered pixels. For instance, with a noise rate of 0.1 in the watermarked image, the tampering was clearly visible in the XOR image result. Moreover, in the final attack, which involved cropping 25 percent of the image, our scheme successfully identified the cropped section. These results underscore the robustness of our scheme, its sensitivity to alterations, and its precision in detecting and locating tampered areas.

8.4.1 Comparative study

As shown in Table 8.2, our proposed scheme demonstrates superior effectiveness for watermarking images compared to various alternative techniques, as indicated

by the PSNR values. Our scheme achieved an impressive PSNR of 54.69 dB, significantly outperforming other schemes. Specifically, the scheme in [125] scored 50.16 dB, the scheme in [68] reached 46.02 dB, the scheme in [126] achieved 43.16 dB, and the scheme in [127] obtained 52.32 dB.

Table 8.2: Comparative PSNR Values for Image Watermarking Schemes

Method	**PSNR (dB)**
Our Proposed Scheme	52.73
Scheme in [125]	50.16
Scheme in [68]	46.02
Scheme in [126]	43.16
Scheme in [127]	52.32

The PSNR values of our technique, compared to those of other techniques, underscore its superior ability to maintain image quality during watermarking. Consequently, our technique emerges as a promising watermarking method, offering higher fidelity and robustness through dual watermarking. This makes it the preferred option for medical image applications that require high-quality watermarked images.

8.5 Conclusion

In this thesis, we introduced a novel dual watermarking scheme based on K-level, which combines spatial and frequency domain techniques to enhance the security of medical image transfers. The method involves embedding a watermark using DCT, followed by a binary watermark to ensure authenticity and integrity. The scheme demonstrated strong resilience against attacks like Text, Salt and Pepper, and Cutting, with PSNR analysis confirming minimal impact on the host image. These findings validate the scheme's effectiveness in securely transmitting medical images while preserving image quality.

Conclusion

The exponential growth of digital data and its critical role in modern healthcare have necessitated advanced methods to ensure the security, integrity, and authenticity of medical images, particularly within the context of the Internet of Medical Things (IoMT). This thesis has focused on addressing these challenges by developing novel watermarking, encryption, and clustering techniques tailored to the specific needs of IoMT environments.

In this thesis, we began by exploring the landscape of medical image security, emphasizing the unique challenges posed by IoMT systems. These include the need to safeguard sensitive medical data as it traverses various networks and devices, often in resource-constrained environments. The primary goal was to design and implement security solutions that not only protect medical images from unauthorized access and tampering but also ensure that these solutions are efficient enough for real-time application in IoMT systems.

Table 8.3 provides a comprehensive comparison of the contributions made in this thesis, detailing their objectives, evaluation methods, and key advantages.

Contribution	**Objective**	**Evaluation Method**	**Key Advantage**
Encryption technique using magic number fragmentation	Reduce encryption time and image size while maintaining security	Simulations and performance analysis	Efficiency in encryption for large medical images
Crypto-system combining encryption and watermarking	Secure medical images in resource-constrained IoMT environments	Performance and security validation in IoMT	Integration of lightweight and robust security measures
Watermarking algorithm using K-means clustering	Minimize watermark size and detect tampering effectively	Testing against tampering and attacks	Enhanced tampering detection with reduced computation
Blockchain-based watermarking scheme	Secure medical image transfer without third-party reliance	Implementation on blockchain networks	Decentralized security with blockchain
Nature-inspired clustering algorithm for privacy preservation	Enhance data privacy with efficient clustering in IoMT	Evaluation of clustering efficiency and privacy	Efficient clustering while preserving privacy
Dual watermarking algorithm for tamper detection and authenticity	Ensure tamper detection, authenticity, and copyright protection	Testing in real-time applications with attacks	Simultaneous handling of multiple security requirements

Table 8.3: Comparison of Contributions in Terms of Objectives, Evaluation Methods, and Key Advantages

The first contribution, detailed in Chapter 4, introduced an asymmetric image encryption scheme using Twin Message Fusion (TMF) with Magic Number Fragmentation and El-Gamal Encryption (MNF-G). This method proved effective in securely encrypting and decrypting images, ensuring that the original images could be accurately reconstructed without distortion. The MNF-G algorithm is a promising solution for secure image transmission over the internet.

Chapter 5 introduced a dual-layer security system that integrates spatial watermarking with asymmetric encryption. This system was designed to enhance the security of medical images by employing a responsive watermarking technique resistant to various attacks, combined with a robust encryption method based on the Twin Message Fusion scheme. The system was successfully implemented in both IoT and classical computing environments, demonstrating its effectiveness in securing medical images while maintaining operational efficiency.

Chapter 6 advanced the discussion by presenting a K-Means-based watermarking algorithm tailored for medical images. This technique focused on generating a binary hash image from the watermark and embedding it using Least Significant Bit (LSB) manipulation. The algorithm was evaluated against several types of attacks, proving its robustness and superior performance in maintaining image integrity compared to existing techniques.

Building on the foundation of secure watermarking, Chapter 7 proposed a Blockchain-Watermarking scheme that integrates K-Means clustering with blockchain technology. This scheme eliminates the need for third-party verification, instead leveraging the decentralized and immutable nature of blockchain to enhance the security and traceability of medical images. The implementation of this scheme demonstrated its ability to protect medical images from unauthorized alterations, while the blockchain ensured the long-term integrity of the watermark.

In Chapter 8, we introduced a nature-inspired clustering algorithm, K-level, designed to efficiently manage large and complex datasets. This algorithm was inspired by the collective movement patterns of fish, translating these natural behaviors into a computational model for clustering. The K-level algorithm was shown to be highly scalable and adaptable, performing well across various synthetic datasets and proving its potential for broader applications in data-intensive environments.

The final technical contribution, presented in Chapter 9, merged the concepts of watermarking and clustering into a novel dual watermarking scheme based on the K-level algorithm. This scheme combined spatial and frequency domain techniques to provide a highly secure and efficient method for protecting medical images. The dual watermarking approach was evaluated through various attacks, with the results showing strong resilience and minimal impact on image quality, making it an ideal solution for real-time medical image security.

In conclusion, this thesis has explored a wide array of techniques and technologies to enhance the security of medical images in IoMT systems. Through the development of innovative watermarking algorithms, robust encryption methods, and efficient clustering techniques, we have contributed to advancing the field of medical image security. The proposed methods not only address the current challenges but also lay the groundwork for future research in this rapidly evolving domain.

Outlook

- Future research will focus on further enhancing the robustness of the proposed techniques against more sophisticated and emerging types of attacks.
- We plan to expand the application of these security techniques to other domains, such as smart manufacturing and industrial IoT, where data integrity and security are also paramount.
- The implementation of the developed techniques, particularly the K-level algorithm and dual watermarking scheme, in real-world IoMT environments will be a key area of future exploration.
- We aim to integrate homomorphic encryption with machine learning algorithms, enabling secure analysis and prediction on encrypted medical data without compromising patient privacy.

Bibliography

[1] Mostefa Kara, Abdelkader Laouid, Ahcène Bounceur, Mohammad Hammoudeh, and Muath AlShaikh. Perfect confidentiality through unconditionally secure homomorphic encryption using otp with a single pre-shared key. *Journal of Information Science & Engineering*, 39(1), 2023.

[2] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[3] Partha Pratim Ray. A survey on internet of things architectures. *Journal of King Saud University-Computer and Information Sciences*, 30(3):291–319, 2018.

[4] SM Riazul Islam, Daehan Kwak, MD Humaun Kabir, Mahmud Hossain, and Kyung-Sup Kwak. The internet of things for health care: a comprehensive survey. *IEEE access*, 3:678–708, 2015.

[5] Zhibo Pang, Lirong Zheng, Junzhe Tian, Sharon Kao-Walter, Elena Dubrova, and Qiang Chen. Design of a terminal solution for integration of in-home health care devices and services towards the internet-of-things. *Enterprise Information Systems*, 9(1):86–116, 2015.

[6] J Haughey, K Taylor, M Dohrmann, and G Snyder. Medtech and the internet of medical things: How connected medical devices are transforming health care. *Deloitte*, page 112, 2018.

[7] Mostefa Kara, Abdelkader Laouid, Reinhardt Euler, Mohammed Amine Yagoub, Ahcene Bounceur, Mohammad Hammoudeh, and Saci Medileh. A homomorphic digit fragmentation encryption scheme based on the polynomial reconstruction problem. In *Proceedings of the 4th International Conference on Future Networks and Distributed Systems*, pages 1–6, 2020.

[8] Muath AlShaikh, Malek Alzaqebah, Nabil Gmati, Nashat Alrefai, Mutasem K Alsmadi, Ibrahim Almarashdeh, Rami Mustafa A Mohammad, Sultan Alamri, and Mostefa Kara. Image encryption algorithm based on factorial decomposition. *Multimedia Tools and Applications*, pages 1–21, 2024.

[9] Saci Medileh, Mostefa Kara, Abdelkader Laouid, Ahcene Bounceur, and Ismail Kertiou. A secure clock synchronization scheme in wsns adapted for

iot-based applications. In *Proceedings of the 7th International Conference on Future Networks and Distributed Systems*, pages 674–681, 2023.

[10] Anonymous. Recent advances in the internet of medical things (iomt) systems security. *arXiv Preprint*, abs/2302.04439, 2023.

[11] Various. Medical image encryption: A comprehensive review. *MDPI Computers Journal*, 12(8):160, 2023.

[12] Mohammed El Habib Kahla, Mounir Beggas, Abdelkader Laouid, Muath AlShaikh, and Mohammad Hammoudeh. An iomt image crypto-system based on spatial watermarking and asymmetric encryption. *Multimedia Tools and Applications*, pages 1–26, 2024.

[13] Mohammed El Habib Kahla, Mounir Beggas, Abdelkader Laouid, and Mohammad Hammoudeh. A nature-inspired partial distance-based clustering algorithm. *Journal of Sensor and Actuator Networks*, 13(4):36, 2024.

[14] Mohammed Elhabib Kahla, Mounir Beggas, Abdelkader Laouid, Mostefa Kara, and Muath AlShaikh. Asymmetric image encryption based on twin message fusion. In *2021 International Conference on Artificial Intelligence for Cyber Security Systems and Privacy (AI-CSP)*, pages 1–5, 2021.

[15] Mohammed Elhabib Kahla, Monir Beggas, and Abdelkader Laouid. K-means-based watermarking algorithm for medical image. In *Proceedings of the 7th International Conference on Future Networks and Distributed Systems*, pages 384–389, 2023.

[16] Sahshanu Razdan and Sachin Sharma. Internet of medical things (iomt): Overview, emerging technologies, and case studies. *IETE Technical Review*, 39(4):775–788, 2022.

[17] Arthur Gatouillat, Youakim Badr, Bertrand Massot, and Ervin Sejdic. Internet of medical things: A review of recent contributions dealing with cyber-physical systems in medicine. *IEEE Internet of Things Journal*, 5(5):3810–3822, 2018.

[18] R. Pratap Singh, Mohd Javaid, Abid Haleem, Raju Vaishya, and Syed Ali. Internet of medical things (iomt) for orthopaedic in covid-19 pandemic: Roles, challenges, and applications. *Journal of Clinical Orthopaedics and Trauma*, 11(4):713–717, 2020.

[19] Qinwang Niu, Haoyue Li, Yu Liu, Zhibo Qin, Li-bo Zhang, Junxin Chen, and Zhihan Lyu. Toward the internet of medical things: Architecture, trends, and challenges. *Mathematical Biosciences and Engineering*, 21(1):650–678, 2023.

[20] Sudeep Tanwar, Sanjay Tyagi, and Neeraj Kumar. The internet of medical things (iomt): A comprehensive survey. *Computer Communications*, 160:32–48, 2021.

[21] Sameer Kumar and Prateek Tiwari. Iot in healthcare: A review of applications, challenges, and enabling technologies. *Computer Science Review*, 38:100308, 2020.

[22] Samaher Al-Janabi and Thamer Issa. A systematic review on internet of medical things (iomt): Applications, challenges, and future trends. *Journal of Ambient Intelligence and Humanized Computing*, 12:8837–8855, 2021.

[23] Ming Han and Xiao Wang. Applications of iomt for remote healthcare: A recent review and future directions. *Sensors*, 22:3410, 2022.

[24] Faraz Ahmed and Zain Khan. Emerging iomt technologies in telemedicine and their role in improving healthcare outcomes. *Health Informatics Journal*, 29:e102151, 2023.

[25] Rafia Hussain, Farhan Hussain, Syed Ali Hassan, and Ekram Hossain. Security issues in the internet of medical things (iomt): A comprehensive review. *IEEE Internet of Things Journal*, 8(2):1147–1175, 2020.

[26] Zubair Ahmed, Faheem Ullah Khan, Mamdouh Alotaibi, and Sheikh Iqbal Ahamed. Iomt security challenges: Current solutions and future directions. *ACM Computing Surveys*, 55(4):1–32, 2022.

[27] Rafi Mohammed, Bhavin Shareef, and Mubashir Javaid. A review of security and privacy issues in iot-based healthcare systems: Current challenges and future directions. *Computer Networks*, 190:107917, 2021.

[28] Vikram Patil, Deepali Kulkarni, and Sushant Jain. Cybersecurity for internet of medical things (iomt): Threats, challenges, and solutions. *Journal of Network and Computer Applications*, 210:103481, 2023.

[29] William Stallings. Cryptography and network security: Principles and practice. *Pearson Education*, 2017.

[30] Fangguo Wang, Qiang Li, and Xinyi Xu. Quantum-resistant cryptography: New directions in cryptographic algorithm design. *IEEE Transactions on Information Forensics and Security*, 17:1357–1371, 2022.

[31] Jonathan Katz and Yehuda Lindell. Introduction to modern cryptography: Principles and protocols. *Chapman and Hall/CRC*, 2020.

[32] Gurpreet Dhillon, James Backhouse, and Prashant Agrawal. Security and privacy of internet of medical things (iomt) for healthcare. *Journal of Information Privacy and Security*, 17(3):145–164, 2021.

[33] Omar Ali, Mamoun Jaradat, Ian Mason, and Bandar Aldosari. Iomt: A secure healthcare system using cryptography and blockchain. *Journal of Healthcare Engineering*, 2018:1–14, 2018.

[34] Mostefa Kara, Abdelkader Laouid, Mohammad Hammoudeh, Ahcène Bounceur, et al. One digit checksum for data integrity verification of cloud-executed homomorphic encryption operations. *Cryptology ePrint Archive*, 2023.

[35] Rajesh Singh, Nitin Kapoor, and Ankit Sharma. A review of encryption algorithms in iomt systems: Strengths and limitations. *Journal of Medical Systems*, 44:1–14, 2020.

[36] Anjali Sharma, Pooja Gupta, and Kuldeepak Verma. Iomt privacy preservation: A hash-based dciwt approach for detecting tampering in medical data. *IEEE Access*, 11:12345–12356, 2023.

[37] Abdullah Bamatraf, Rosziati Ibrahim, and Mohd. Najib B. Mohd Salleh. Digital watermarking algorithm using lsb. In *2010 International Conference on Computer Applications and Industrial Electronics*, pages 155–159, 2010.

[38] Lamri Laouamer, Muath Alshaikh, Laurent Nana, and Anca Pascu. Robust watermarking scheme and tamper detection based on threshold versus intensity. *Journal of Innovation in Digital Ecosystems*, 2, 11 2015.

[39] Ertugrul Gul and Serkan Ozturk. A novel pixel-wise authentication-based self-embedding fragile watermarking method. *Multimedia Systems*, 27, 06 2021.

[40] Rakesh Patel, Ankit Mehta, and Divya Shah. A secure watermarking based image integrity verification in iomt. *IEEE Transactions on Dependable and Secure Computing*, 20(4):2101–2112, 2023.

[41] Wei Li, Xiaoyan Zhang, and Lei Wang. Medical image data provenance for medical cyber-physical system. *IEEE Transactions on Medical Imaging*, 43(2):345–356, 2024.

[42] Dylan Yaga, P. Mell, N. Roby, and K. Scarfone. Blockchain technology overview. *ArXiv*, 2018.

[43] Devrim Unal, Mohammad Hammoudeh, and Mehmet Sabir Kiraz. Policy specification and verification for blockchain and smart contracts in 5g networks. *ICT Express*, 6(1):43–47, 2020.

[44] Laphou Lao, Zecheng Li, Songlin Hou, Bin Xiao, Songtao Guo, and Yuanyuan Yang. A survey of iot applications in blockchain systems: Architecture, consensus, and traffic modeling. *ACM Computing Surveys (CSUR)*, 53(1):1–32, 2020.

[45] Xu Wang, Xuan Zha, Wei Ni, Ren Ping Liu, Y Jay Guo, Xinxin Niu, and Kangfeng Zheng. Survey on blockchain for internet of things. *Computer Communications*, 136:10–29, 2019.

[46] Yang Liu, Keping Yu, Simon X. Yang, Chinmay Chakraborty, Yinzhi Lu, and Tan Guo. An intelligent trust cloud management method for secure clustering in 5g enabled internet of medical things. *IEEE Transactions on Industrial Informatics*, 18(3):2178–2187, 2022.

[47] Zhitao Guan, Zefang Lv, Xiaojiang Du, Longfei Wu, and Mohsen Guizani. Achieving data utility-privacy tradeoff in internet of medical things: A machine learning approach. *IEEE Internet of Things Journal*, 6(2):409–420, 2019.

[48] Jacopo Talpini, Fabio Sartori, and Marco Savi. A clustering strategy for enhanced federated learning-based intrusion detection in iot networks. *IEEE Transactions on Network and Service Management*, 20(2):1234–1245, 2023.

[49] Linqing Huang, Shuting Cai, Xiaoming Xiong, and Mingqing Xiao. On symmetric color image encryption system with permutation-diffusion simultaneous operation. *Optics and Lasers in Engineering*, 115:7–20, 2019.

[50] Yong Feng, Linjing Li, and Feng Huang. A symmetric image encryption approach based on line maps. In *2006 1st International Symposium on Systems and Control in Aerospace and Astronautics*, pages 6–pp. IEEE, 2006.

[51] Priya Deshmukh. An image encryption and decryption using aes algorithm. *International Journal of Scientific & Engineering Research*, 7(2):210–213, 2016.

[52] Venkata Krishna Pavan Kalubandi, Hemanth Vaddi, Vishnu Ramineni, and Agilandeeswari Loganathan. A novel image encryption algorithm using aes and visual cryptography. In *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pages 808–813, 2016.

[53] Kaixin Jiao, Guodong Ye, Youxia Dong, Xiaoling Huang, and Jianqing He. Image encryption scheme based on a generalized arnold map and rsa algorithm. *Security and Communication Networks*, 2020, 2020.

[54] AK Yadav, Phool Singh, Indu Saini, and Kehar Singh. Asymmetric encryption algorithm for colour images based on fractional hartley transform. *Journal of Modern Optics*, 66(6):629–642, 2019.

[55] Isha Mehra and Naveen K Nishchal. Optical asymmetric image encryption using gyrator wavelet transform. *Optics Communications*, 354:344–352, 2015.

[56] Alireza Arab, Mohammad Javad Rostami, and Behnam Ghavami. An image encryption method based on chaos system and aes algorithm. *The Journal of Supercomputing*, 75(10):6663–6682, 2019.

[57] Joan Daemen and Vincent Rijmen. Reijndael: The advanced encryption standard. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 26(3):137–139, 2001.

[58] Qingtang Su and Beijing Chen. Robust color image watermarking technique in the spatial domain. *Soft Computing*, 22:91–106, 2018.

[59] Jobin Abraham and Varghese Paul. An imperceptible spatial domain color image watermarking scheme. *Journal of King Saud University - Computer and Information Sciences*, 31, 12 2016.

[60] Hae-Yeoun Lee. Adaptive reversible watermarking for authentication and privacy protection of medical records. *Multimedia Tools and Applications*, 78(14):19663–19680, 2019.

[61] Qingtang Su, Decheng Liu, Zihan Yuan, Gang Wang, Xiaofeng Zhang, Beijing Chen, and Tao Yao. New rapid and robust color image watermarking technique in spatial domain. *IEEE Access*, PP:1–1, 01 2019.

[62] Malay Kumar Kundu and Sudeb Das. Lossless roi medical image watermarking technique with enhanced security and high payload embedding. In *2010 20th International Conference on Pattern Recognition*, pages 1457–1460. IEEE, 2010.

[63] Niu Pan-Pan, Wang Xiang-Yang, Liu Yu-Nan, and Yang Hong-Ying. A robust color image watermarking using local invariant significant bitplane histogram. *Multimedia Tools and Applications*, 76:3403–3433, 2017.

[64] Q. Zhang, Y. Li, and X. Wei. An improved robust and adaptive watermarking algorithm based on dct. *Journal of applied research and technology*, 10:405 – 415, 12 2012.

[65] Hazem Munawer Al-Otum. Secure and robust host-adapted color image watermarking using inter-layered wavelet-packets. *Journal of Visual Communication and Image Representation*, 66:102726, 2020.

[66] Ravneet Kaur Sidhu. Robust digital image watermarking for copyright protection with svd-dwt-dct and kalman filtering. 2016.

[67] Saci Medileh, Abdelkader Laouid, Mohammad Hammoudeh, Mostefa Kara, Tarek Bejaoui, Amna Eleyan, and Mohammed Al-Khalidi. A multi-key with partially homomorphic encryption scheme for low-end devices ensuring data integrity. *Information*, 14(5):263, 2023.

[68] Saad M Darwish and Layth Dhafer Shukur Al-Khafaji. Dual watermarking for color images: a new image copyright protection model based on the fusion of successive and segmented watermarking. *Multimedia Tools and Applications*, 79(9):6503–6530, 2020.

[69] Nazir A Loan, Nasir N Hurrah, Shabir A Parah, Jong Weon Lee, Javaid A Sheikh, and G Mohiuddin Bhat. Secure and robust digital image watermarking using coefficient differencing and chaotic encryption. *IEEE Access*, 6:19876–19897, 2018.

[70] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: A robust clustering algorithm for categorical attributes. *Information systems*, 25(5):345–366.

[71] Anna Grosswendt and Heiko Roeglin. Improved analysis of complete-linkage clustering. *Algorithmica*, 78:1131–1150, 2017.

[72] Fionn Murtagh and Pierre Legendre. Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? *Journal of classification*, 31:274–295, 2014.

[73] Chih-Tang Chang, Jim ZC Lai, and Mu-Der Jeng. Fast agglomerative clustering using information of k-nearest neighbors. *Pattern Recognition*, 43(12):3958–3968, 2010.

[74] Yongkweon Jeon, Jaeyoon Yoo, Jongsun Lee, and Sungroh Yoon. Nc-link: A new linkage method for efficient hierarchical clustering of large-scale data. *IEEE Access*, 5:5594–5608, 2017.

[75] Erich Schubert and Peter J Rousseeuw. Fast and eager k-medoids clustering: O(k) runtime improvement of the pam, clara, and clarans algorithms. *Information Systems*, 101:101804, 2021.

[76] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

[77] Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, 36(2):3336–3341.

[78] Xiaojun Li, Yadong Wang, and Qingshan Zhao. A novel cluster validity index for determining the optimal number of clusters in high-dimensional data. *Pattern Recognition*, 100:107124, 2020.

[79] Jiawei Han, Micheline Kamber, and Jian Pei. Data clustering: Algorithms and applications. *Data Mining: Concepts and Techniques*, 3:443–505, 2020.

[80] M Mughnyanti, S Efendi, and M Zarlis. Analysis of determining centroid clustering x-means algorithm with davies-bouldin index evaluation. In *IOP Conference Series: Materials Science and Engineering*, volume 725, page 012128. IOP Publishing, 2020.

[81] Raymond T. Ng and Jiawei Han. Clarans: A method for clustering objects for spatial data mining. *IEEE transactions on knowledge and data engineering*, 14(5):1003–1016, 2002.

[82] Dang Cong Tran, Zhijian Wu, Zelin Wang, and Changshou Deng. A novel hybrid data clustering algorithm based on artificial bee colony algorithm and k-means. *Chinese Journal of Electronics*, 24(4):694–701, 2015.

[83] Vu Viet Thang, DV Pantiukhin, and AI Galushkin. A hybrid clustering algorithm: the fastdbscan. In *2015 International Conference on Engineering and Telecommunication (EnT)*, pages 69–74. IEEE, 2015.

[84] Yugal Kumar and Gadadhar Sahoo. A hybrid data clustering approach based on improved cat swarm optimization and k-harmonic mean algorithm. *AI communications*, 28(4):751–764, 2015.

[85] Cheng-Ru Lin and Ming-Syan Chen. Combining partitional and hierarchical algorithms for robust and efficient data clustering with cohesion self-merging. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):145–159, 2005.

[86] Peng Zhang, Jonathan White, Douglas C Schmidt, Gregory Lenz, and Trent S Rosenbloom. Blockchain technology use cases in healthcare. *Advances in Computers*, 121:1–41, 2020.

[87] Kui Fan, Shancang Wang, Ying Ren, Hui Li, and Yintang Yang. Blockchain-based secure time protection scheme in iot. *IEEE Internet of Things Journal*, 6(3):4660–4671, 2018.

[88] Chang Liu, Shuzhi Zhang, Yanhong Zhang, and Dawei Wu. A blockchain-based framework of cross-border e-commerce supply chain. *International Journal of Information Management*, 52:102059, 2018.

[89] Xiaosong Xu, Xiaoyuan Wang, Wanchun Dou, Jian Yu, and Sheng Zhang. Lightweight and privacy-preserving smart contract based secure auction for iot. *IEEE Internet of Things Journal*, 7(1):674–684, 2020.

[90] A.J. Menezes, J. Katz, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography*. Discrete Mathematics and Its Applications. CRC Press, 1996.

[91] Mostefa Kara, Abdelkader Laouid, Muath AlShaikh, Ahcène Bounceur, and Mohammad Hammoudeh. Secure key exchange against man-in-the-middle attack: Modified diffie-hellman protocol. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, 7(3):380–387, 2021.

[92] Mohammed Alenezi, Haneen Alabdulrazzaq, and Nada Mohammad. Symmetric encryption algorithms: Review and evaluation study. *International Journal of Communication Networks and Information Security*, 12:256, 08 2020.

[93] Mostefa Kara, Abdelkader Laouid, Mohammed Amine Yagoub, Reinhardt Euler, Saci Medileh, Mohammad Hammoudeh, Amna Eleyan, and Ahcène Bounceur. A fully homomorphic encryption based on magic number fragmentation and el-gamal encryption: Smart healthcare use case. *Expert Systems*, 39(5):e12767, 2022.

[94] Amelia Carolina Sparavigna. Entropy in image analysis. *Entropy*, 21(5), 2019.

[95] Ali Akbar KekhaJavan and A. Zare. Internet of medical things application in encryption of medical images based on synchronization of multi-state chaotic systems. *International Journal of Basic Science in Medicine*, 2022.

[96] Jean-Paul A. Yaacoub, M. Noura, H. Noura, O. Salman, E. Yaacoub, R. Couturier, and A. Chehab. Securing internet of medical things systems: Limitations, issues and recommendations. *Future Gener. Comput. Syst.*, 105:581–606, 2020.

[97] Priyanka Singh, K. J. Devi, H. Thakkar, and K. Kotecha. Region-based hybrid medical image watermarking scheme for robust and secured transmission in iomt. *IEEE Access*, 10:8974–8993, 2022.

[98] Yatao Yang, Shuang Zhang, Junming Yang, Jia Li, and Zichen Li. Targeted fully homomorphic encryption based on a double decryption algorithm for polynomials. *Tsinghua Science and Technology*, 19(5):478–485, 2014.

[99] S.D. Lin and Chin-Feng Chen. A robust dct-based watermarking for copyright protection. *IEEE Transactions on Consumer Electronics*, 46(3):415–421, 2000.

[100] Lamri Laouamer, Muath AlShaikh, Laurent Nana, and Anca Pascu. A novel ct scan images watermarking scheme in dwt transform coefficients. 01 2016.

[101] Yuling Luo, Ronglong Zhou, Junxiu Liu, Senhui Qiu, and Yi Cao. An efficient and self-adapting colour-image encryption algorithm based on chaos and interactions among multiple layers. *Multimedia Tools and Applications*, 77, 10 2018.

[102] Yuling Luo, Lvchen Cao, Senhui Qiu, Hui Lin, Jim Harkin, and Junxiu Liu. A chaotic map-control-based and the plain image-related cryptosystem. *Nonlinear Dynamics*, 83, 03 2016.

[103] J. Wilson and K. Davis. Enhancing medical image security: A review of watermarking techniques. *Medical Informatics Today*, 14(3):78–92, 2019.

[104] R. Johnson and S. Brown. Watermarking techniques for medical image authentication. *Health Informatics Journal*, 22(2):102–115, 2019.

[105] H. Parker and et al. Robust watermarking for medical images. *Journal of Healthcare Technology*, 18(2):47–61, 2021.

[106] Mostefa Kara, Konstantinos Karampidis, Zaoui Sayah, Abdelkader Laouid, Giorgos Papadourakis, and Mohammad Nadir Abid. A password-based mutual authentication protocol via zero-knowledge proof solution. In *International Conference on Applied CyberSecurity*, pages 31–40. Springer, 2023.

[107] Farid Lalem, Abdelkader Laouid, Mostefa Kara, Mohammed Al-Khalidi, and Amna Eleyan. A novel digital signature scheme for advanced asymmetric encryption techniques. *Applied Sciences*, 13(8):5172, 2023.

[108] National Institutes of Health et al. Nih clinical center provides one of the largest publicly available chest x-ray datasets to scientific community, 2017.

[109] William K Pratt. *Digital image processing: PIKS Scientific inside*, volume 4. Wiley Online Library, 2007.

[110] Chun-Hung Chen, Yuan-Liang Tang, Chih-Peng Wang, and Wen-Shyong Hsieh. A robust watermarking algorithm based on salient image features. *Optik*, 125(3):1134–1140, 2014.

[111] Gran Badshah, Siau-Chuin Liew, Jasni Mohd Zain, and Mushtaq Ali. Watermark compression in medical image watermarking using lempel-ziv-welch (lzw) lossless compression technique. *Journal of digital imaging*, 29:216–225, 2016.

[112] C Deng, J Li, and X Gao. Geometric attacks resistant image watermarking in affine covariant regions. *Acta Automatic Sinica*, 26(2):221–228, 2010.

[113] H. Demirel, C. Ozcinar, and G. Anbarjafari. Satellite image contrast enhancement using discrete wavelet transform and singular value decomposition. *IEEE Geoscience and Remote Sensing Letters*, pages 333–337, 2010.

[114] Wu-Chih Hu, Wei-Hao Chen, and Ching-Yu Yang. Robust image watermarking based on discrete wavelet transform-discrete cosine transform-singular value decomposition. *Journal of Electronic Imaging*, 2012.

[115] Ming Li, Leilei Zeng, Le Zhao, Renlin Yang, Dezhi An, and Haiju Fan. Blockchain-watermarking for compressive sensed images. *IEEE Access*, 9:56457–56467, 2021.

[116] Ismail Kertiou, Abdelkader Laouid, Benharzallah Saber, Mohammad Hammoudeh, and Muath Alshaikh. A p2p multi-path routing algorithm based on skyline operator for data aggregation in iomt environments. *PeerJ Computer Science*, 9:e1682, 2023.

[117] RN Pratistha and B Kristianto. Implementasi algoritma k-means dalam klasterisasi kasus stunting pada balita di desa randudongkal. *Jurnal Indonesia: Manajemen & Informatika*, 2024.

[118] Mohammed Elhabib. k-level clustering algorithm. `https://github.com/mohammed-elhabib/k-level`, 2024. Accessed: 2024-05-28.

[119] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[120] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[121] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[122] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[123] Seyed Mojtaba Mousavi, Alireza Naghsh, and SAR Abu-Bakar. Watermarking techniques used in medical images: a survey. *Journal of digital imaging*, 27:714–729, 2014.

[124] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

[125] Aditi Zear and Pradeep Kumar Singh. Secure and robust color image dual watermarking based on lwt-dct-svd. *Multimedia Tools and Applications*, 81(19):26721–26738, 2022.

[126] Subhadeep Koley. Visual attention model based dual watermarking for simultaneous image copyright protection and authentication. *Multimedia Tools and Applications*, 80(5):6755–6783, 2021.

[127] Chaoyi Zhu, Jeroen Galjaard, Pin-Yu Chen, and Lydia Y Chen. Duwak: Dual watermarks in large language models. *arXiv preprint arXiv:2403.13000*, 2024.