PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY OF ECHAHID HAMMA LAKHDAR EL-OUED

Faculty of Exact Science
Department of Computer Science

Thesis

Submitted in partial fulfillment of the requirements for the Doctorate degree
Option: Advanced Information Systems
Presented by **KEBACHE ROMAISSA**

Thesis Title:

The adaptation of data and communication security techniques in the context of the Internet of Things

Jury Members:

President	Abbas Messaoud	MCA	University of El Oued
Supervisor	Laouid Abdelkader	Prof	University of El Oued
Co-Supervisor	Eleyan Amna	MCA	University of Manchester
Examiner	Kahloul Laid	Prof	University of Biskra
Examiner	Beggas Mounir	MCA	University of El Oued
Examiner	Sebaa Abderrazak	MCA	ESTIN- Bejaia

February 2025

Acknowledgments

First and foremost, I extend my deepest gratitude to Allah for granting me the health, strength, perseverance, and unwavering faith necessary to complete this work. I am profoundly grateful to my project supervisor, Dr. Laouid Abdelkader, Professor at Echahid Hamma Lakhdar University, El oued, whose invaluable guidance, insightful suggestions, and unwavering confidence in my abilities greatly contributed to the successful completion of this project. His support in providing a conducive working environment and high-quality supervision was indispensable.

My heartfelt thanks also go to Dr. Kara Mostefa, MCA, at the Scientific and Technology Hub of Sidi Abdellah, Algiers, for his constant support and guidance throughout the duration of this work. I express my deepest appreciation for his continuous encouragement, insightful advice, and the confidence he has consistently shown in me.

I extend my sincere gratitude to Professor Ahcene Bounceur, HDR at the University of Brest, and Professor Mohammad Hammoudeh, PhD at Manchester Metropolitan University for their precious assistance and keen interest in my work. Their expert guidance and support in reviewing and drafting our manuscript have been instrumental in successfully completing this work.

I want to take a moment to express my sincere appreciation to my wonderful friend and respected colleague, Dr. Assia Brek, who serves as an Assistant Professor at Larbi Tebessi University in Tebessa. Her consistent support, insightful feedback, and engaging discussions have been instrumental in advancing my PhD research. I am genuinely thankful for her willingness to share her knowledge and expertise

I am equally grateful to all jury members, for their thoughtful engagement with my work, their willingness, and for doing me the great honor of presiding and serving as a member of the jury to assess and evaluate my efforts in my graduation project. Furthermore, I would like to acknowledge with deep appreciation all the faculty members of the Department of Exact Science at Echahid Hamma Lakhdar University, El Oued. Their dedication to my education, their expertise, and above all, their humbleness, have profoundly enriched my academic journey.

Lastly, I extend my heartfelt thanks to my family, friends, and all those who, in any capacity, contributed to the successful realization of this work. Their unwavering support and encouragement have been invaluable.

Dedication

First of all, I thank Allah, our creator, for giving me the strength, the will and the courage to accomplish this modest work.
To my parents, this work is the result of your endless ambitions and motivations.
I dedicate this work to my father, whom I thank enormously for his efforts, his advice and his superintendence, for all his endless sacrifices and patience.
To my mother, the source of tenderness, the light that guides my paths and leads me to success, for all her assistance and presence in my life.
To my supervisor, Dr. Laouid Abdelkader, for your unwavering patience, guidance, and support throughout these five challenging years. Despite my difficult circumstances, you believed in me when I struggled to believe in myself. Your mentorship has been instrumental in both my academic and personal growth.
To my husband, your wisdom, love, and endless support have been the foundation of my achievements.
To my little angel, my precious daughter Ritel, you are the true future treasure and the flower that brings me hope and joy every day.
To my wing-women and dearest sister Afaf, thank you for your valuable time and support.
To my lovely younger sister Serine, thank you for your patience and being sagaciously beside me.
To all my family members,friends, and colleagues for your continuous support and encouragement.
Finally, To all my teachers without exception.
Thanks,

Abstract

With the continuous expansion of IoT applications, ensuring robust data security has become crucial, particularly in sensitive domains such as healthcare, smart cities, and industrial automation. Traditional security mechanisms are often insufficient due to the heterogeneity and constraints of IoT devices. This research investigates how conventional cryptographic techniques can be optimized for lightweight and efficient implementation in IoT systems. By leveraging homomorphic encryption and adaptive security measures, this work aims to provide a practical and scalable security solution tailored to the constraints of IoT environments.

The Internet of Things (IoT) presents vast opportunities but also significant security challenges due to its interconnected nature. This thesis explores the adaptation of data protection and secure communication techniques to the IoT context. It introduces a novel encryption scheme, Multi-Key Embedded Encryption (MKEE), combined with the Modular Multiplicative Inverse (MMI) technique, achieving a 94% reduction in encrypted data storage. Furthermore, this research presents a deep learning-based model for Tuberculosis detection, demonstrating high accuracy using advanced convolutional neural networks, reaching 99.66% for training and 99.78% for validation.

The experimental results validate the proposed methodologies through real-world case studies, showing significant improvements in both storage efficiency and security robustness. The deep learning model developed for medical image analysis shows promise in enhancing early disease detection. Ultimately, this thesis contributes valuable insights into securing IoT ecosystems while maintaining system performance and usability.

Keywords: Internet of Things, Security, Encryption, Homomorphic Encryption, Deep Learning, Healthcare, Data Protection, Secure Communication.

Résumé

Avec l'expansion continue des applications IoT, garantir la sécurité des données est devenu un enjeu primordial, en particulier dans les domaines sensibles tels que la santé, les villes intelligentes et l'automatisation industrielle. Les mécanismes de sécurité traditionnels se révèlent souvent insuffisants en raison de l'hétérogénéité et des contraintes des dispositifs IoT. Cette recherche explore comment les techniques cryptographiques classiques peuvent être optimisées pour une implémentation légère et efficace dans les systèmes IoT. En exploitant le chiffrement homomorphe et des mesures de sécurité adaptatives, ce travail vise à fournir une solution de sécurité pratique et évolutive, adaptée aux environnements IoT.

L'Internet des objets (IoT) offre de nombreuses opportunités mais soulève également d'importants défis de sécurité en raison de son interconnectivité. Cette thèse explore l'adaptation des techniques de protection des données et de communication sécurisée dans le contexte de l'IoT. Elle propose un schéma de chiffrement novateur, le Multi-Key Embedded Encryption (MKEE), associé à la technique de l'Inverse Multiplicatif Modulaire (MMI), permettant une réduction de 94% de l'espace de stockage des données chiffrées. De plus, cette recherche présente un modèle d'apprentissage profond pour la détection de la tuberculose, atteignant une précision élevée grâce aux réseaux neuronaux convolutifs avancés.

Les résultats expérimentaux obtenus valident les méthodologies proposées à travers des études de cas réelles. Les techniques de chiffrement développées démontrent une amélioration significative de l'efficacité du stockage et de la robustesse de la sécurité. Le modèle d'apprentissage profond pour l'analyse des images médicales montre également un potentiel considérable pour améliorer la détection précoce des maladies, ouvrant ainsi la voie à une transformation de la sécurité numérique dans le domaine de la santé. Cette thèse apporte ainsi une contribution précieuse à la sécurisation des écosystèmes IoT tout en préservant la performance et l'accessibilité des systèmes.

Mots clés: Internet des objets, Sécurité, Chiffrement, Chiffrement homomorphe, Apprentissage profond, Santé, Protection de Données, Communication Sécurisée.

ملخـص

مع التوسع المستمر في تطبيقات إنترنت الأشياء، أصبح ضمان أمان البيانات أمرًا بالغ الأهمية، لا سيما في المجالات الحساسة مثل الرعاية الصحية، والمدن الذكية، والأتمتة الصناعية. غالبًا ما تكون الآليات الأمنية التقليدية غير كافية بسبب تباين الأجهزة وتقيداتها في بيئة إنترنت الأشياء. تستكشف هذه الدراسة كيفية تحسين التقنيات التشفيرية التقليدية لتكون أكثر كفاءة وخفة في التنفيذ ضمن أنظمة إنترنت الأشياء. ومن خلال الاستفادة من التشفير المتجانس وتدابير الأمان التكيفية، يهدف هذا البحث إلى تقديم حل أمني عملي وقابل للتوسع يلائم احتياجات بيئات إنترنت الأشياء.

يوفر إنترنت الأشياء (IoT) فرصًا كبيرة، لكنه يطرح أيضًا تحديات أمنية خطيرة بسبب طبيعته المترابطة. تستكشف هذه الأطروحة تكييف تقنيات حماية البيانات والاتصالات الآمنة في سياق إنترنت الأشياء. ويتم تقديم مخطط تشفير جديد، وهو التشفير المتعدد المفاتيح ،(MKEE) مع تقنية المعكوس الضربي المعياري ،(MMI) مما يحقق تقليلًا بنسبة 94٪ في مساحة تخزين البيانات المشفرة. علاوة على ذلك، تقدم هذه الدراسة نموذجًا يعتمد على التعلم العميق للكشف عن مرض السل، محققًا دقة عالية باستخدام الشبكات العصبية الالتفافية المتقدمة.

تؤكد النتائج التجريبية صحة المنهجيات المقترحة من خلال دراسات حالة واقعية، حيث أظهرت تقنيات التشفير المطورة تحسينات كبيرة في كفاءة التخزين وقوة الأمان. كما يُظهر نموذج التعلم العميق المستخدم في تحليل الصور الطبية قدرة كبيرة على تحسين الكشف المبكر عن الأمراض، مما قد يُحدث نقلة نوعية في أمن الرعاية الصحية الرقمية. وتوفر هذه الأطروحة رؤية متعمقة لتأمين أنظمة إنترنت الأشياء مع الحفاظ على الأداء العالي وسهولة الاستخدام.

الكلمات المفتاحية: إنترنت الأشياء، الأمن، التشفير, التشفير المتجانس، التعلم العميق, الرعاية الصحية , حماية البيانات, التواصل المأمن.

Contents

List of Figures

List of Tables

List of Abbreviations

ABAC Attribute-Based Access Control. 7

ACS Access control systems. 7

AES Advanced Encryption Standard. 6, 28

AI Artificial Intelligence. 6

ANOVA analysis of variance. 19

BD Big Data. 6

CNNs Convolutional neural networks. 8

CRUSAP CRO-Universal Security Authentication Protocol. 32

DDoS Distributed Denial of Service. 16

DES Data Encryption Standard. 28

DL Deep-Learning. 7

DLP discrete logarithm problem. 17

DS Distributed Systems. 2

ECC Elliptic Curve Cryptography. 6

ENC Encryption. 6

ES Embedded Systems. 4

ESRAS Enhanced Security Rank Authentication Scheme. 32

FHE Fully Homomorphic Encryption. 38

HE Homomorphic Encryption. 7

IDEA International Data Encryption Algorithm. 28

IDS Intrusion Detection Systems. 7

IIoT Industrial Internet of Things. 11

IoMT Internet of Military Things. 11

IoT Internet of Things. 2, 5

KMS Key Management Systems. 32

MD4 Message-Digest Algorithm. 30

MKEE Multi-Key Embedded Encryption. 7

MMI Modular Multiplicative Inverse. 7

NFC Near Field Communications. 13

NIST National Institute of Standards and Technology. 25

OKMS Oblivious Key Management Systems. 33

OPRF Oblivious Pseudorandom Functions. 33

PKC Public key cryptography. 27

PKI Public Key Infrastructure. 7

PQC Post-quantum cryptography. 33

RBAC Role-Based Access Control. 7

RFID Radio-Frequency IDentification. 13

RNNs Recurrent Neural Networks. 47

RSA Rivest–Shamir–Adleman. 16

SHA-1 Secure Hash Algorithm 1. 30

SKC Symmetric key cryptography. 28

SSL/TLS Secure Sockets Layer/Transport Layer Security. 31

SWHE Somewhat Homomorphic Encryption. 38

TB Tuberculosis. 7

WHO World Health Organization. 11

Chapter 1

Introduction

1.1 Generalities

The Internet of Things (IoT) is revolutionizing the way we interact with technology, blending the digital and physical worlds in unprecedented ways. By enabling the interconnection of everyday objects and devices over the internet, IoT facilitates seamless communication and data exchange, leading to enhanced efficiency, automation, and convenience across various sectors, including healthcare, smart cities, industrial automation, and more. However, the expansive connectivity and data flow inherent in IoT also brings significant security challenges. The diversity and sheer number of IoT devices, along with the continuous transmission of sensitive data, create numerous potential vulnerabilities that can be exploited by malicious actors. Thus, there is a pressing need to adapt and develop robust data and communication security techniques tailored to the unique requirements of IoT environments. This thesis aims to explore and address these challenges by proposing innovative approaches to enhance the security of IoT systems.

Distributed Systems (DS) form the backbone of modern computing infrastructure, consisting of multiple autonomous computing devices that collaborate to achieve common objectives. These systems are characterized by their ability to operate seamlessly across different geographical locations, providing benefits such as enhanced performance, scalability, and fault tolerance. DSs enable efficient resource allocation, real-time data processing, and dynamic system reconfiguration, all of which are crucial for the effective functioning of IoT applications [1] .

In the context of IoT, distributed systems are essential for managing the vast networks of interconnected devices that continuously generate and exchange data. However, the distributed nature of IoT, as shown in Figure 1.1 also introduces significant security challenges. Each node within a DSs can potentially become a point of vulnerability, exposing the entire network to attacks [2].

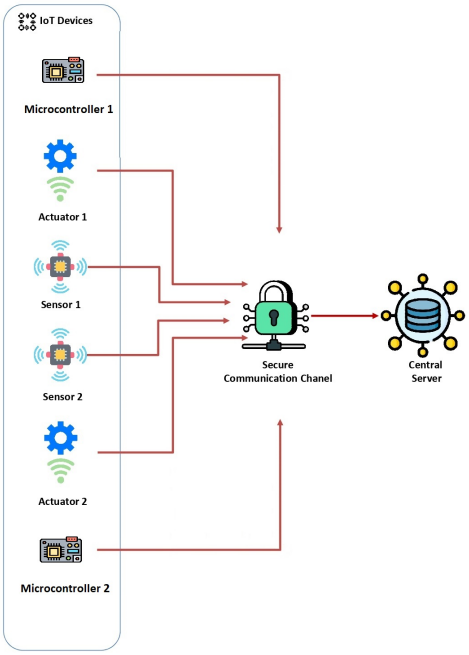

Figure 1.1: A simplified diagram of a distributed IoT system

Ensuring the security of data and communication within distributed IoT systems necessitates the implementation of advanced encryption algorithms, secure communication protocols, and robust authentication mechanisms. Additionally, the heterogeneity of IoT devices, ranging from simple sensors to complex computing units, requires the adaptation of security techniques to address varying device capabilities and constraints. This thesis will delve into strategies for enhancing the security of distributed IoT systems, focusing on the adaptation of existing security techniques and the development of novel approaches.

1.1.1 Distributed systems

DSs are very important in today's technology world, especially with the rise of the Internet of Things (IoT). A distributed system is made up of many connected computers that work together to reach a shared goal. Each computer in this system works on its own, but when they come together, they create a strong network that can handle more tasks, keep working even if some parts fail, and perform better than older, single-computer systems [1].

The significance of distributed systems has increased greatly due to the widespread use of IoT devices. As more devices are added to an IoT network, the need for

systems that can manage large amounts of data and real-time communication also grows. Distributed systems are naturally well-suited for these situations because they can process data close to where it is generated, which lowers delays and prevents slowdowns [3]. This capability to spread out the processing is very important, especially when IoT applications operate across various locations that require quick responses.

Additionally, the fault tolerance provided by distributed systems is very important. In IoT networks, problems can happen at different places due to hardware issues, communication problems, or even cyberattacks. A well-designed distributed system can make sure that problems in one part do not cause the whole system to stop working. By using methods like redundancy and replication, distributed systems can keep working smoothly even if some parts fail [4]. This kind of reliability is crucial for important IoT applications, like smart cities, healthcare, and industrial automation, where stopping can cause serious problems.

Besides scalability and fault tolerance, distributed systems provide better security features. By dividing data among many nodes and using decentralized methods, these systems can reduce the effects of a security breach. For example, blockchain technology, a type of distributed ledger system, is often suggested as a way to improve security in IoT settings [5]. Blockchain helps ensure data accuracy, offers transparency, and stops unauthorized changes, which are all important for protecting communication and data in IoT systems.

Despite their benefits, distributed systems come with challenges. The variety of devices and the constantly changing nature of IoT environments can make it hard to coordinate and sync up different parts of the system [6]. Also, keeping a large, spread-out network secure is complicated because it involves dealing with many different threats that could take advantage of weak spots in communication, devices, or software. Solving these issues is important to make sure that distributed systems can properly support IoT applications without hurting performance or security.

Distributed systems are crucial for the Internet of Things (IoT). They allow for scalable, reliable, and secure operations, making them the foundation of current IoT setups. As IoT keeps growing, the need for strong distributed systems that can handle new security risks and performance needs will become even more important.

1.1.2 Embedded Systems

Embedded Systems (ES) are specialized computing systems designed to perform dedicated functions within larger mechanical or electrical systems. These systems provide the processing power and control necessary for IoT devices to operate

effectively. ES are typically characterized by their efficiency, reliability, and real-time operational capabilities, often functioning with limited resources [7].

The integration of ES into IoT devices introduces unique security challenges. Due to their constrained nature, embedded systems often lack the computational power and memory required to implement robust security measures found in traditional computing environments. This limitation makes them particularly vulnerable to attacks such as side-channel attacks, firmware tampering, and physical attacks [8]. Addressing these challenges requires adapting security techniques to the specific constraints and requirements of embedded systems. This includes developing lightweight encryption algorithms, secure boot processes, and efficient authentication protocols. This thesis will investigate how existing data and communication security techniques can be adapted and optimized for embedded systems within IoT, ensuring robust protection without compromising performance.

1.1.3 The Internet Of Things

IoT refers to the vast network of physical objects "things" embedded with sensors, software, and other technologies to connect and exchange data with other devices and systems over the Internet. This interconnected ecosystem transforms everyday objects into intelligent entities capable of collecting and sharing data autonomously, leading to significant advancements in various sectors [9].

The significance of IoT lies in its potential to drive innovation, improve efficiency, and create new business opportunities. For example, in healthcare, IoT devices can monitor patient vitals in real-time, enabling prompt medical intervention and improving patient outcomes. In smart cities, IoT can optimize traffic flow, reduce energy consumption, and enhance public safety by integrating various data sources and providing real-time insights [10]. The ability of IoT to seamlessly integrate the physical and digital worlds enhances operational capabilities and provides unprecedented opportunities for automation and data-driven decision-making.

However, the widespread adoption of IoT also raises significant security concerns. The vast amount of data generated by IoT devices and the need for continuous communication create numerous opportunities for cyber threats. Ensuring the confidentiality, integrity, and availability of IoT data is paramount to realizing the full benefits of IoT applications [11]. This thesis will examine the specific security challenges associated with IoT and propose adaptations to data and communication security techniques to effectively address these challenges.

1.1.4 AI and Big Data

Artificial Intelligence (AI) and Big Data (BD) are integral to unlocking the full potential of IoT. AI techniques, such as machine learning and deep learning, enable IoT devices to analyze vast amounts of data, detect patterns, and make informed decisions autonomously. These capabilities are essential for applications like predictive maintenance, where AI can identify potential equipment failures before they occur, and personalized healthcare, where AI can tailor treatments based on individual patient data [12].

BD technologies facilitate the storage, processing, and analysis of the enormous volumes of data generated by IoT devices. They provide the necessary infrastructure for managing data efficiently, enabling real-time analytics and decision-making. The combination of AI and BD in IoT creates powerful systems capable of delivering intelligent insights and automating complex processes [13].

However, the integration of AI and BD with IoT also introduces new security challenges. Protecting sensitive data, ensuring the integrity of AI algorithms, and defending against adversarial attacks are critical considerations [14]. Additionally, the distributed nature of IoT systems and the real-time requirements of many applications demand security solutions that are both robust and efficient. This thesis will explore the intersection of AI, BD, and IoT security, proposing adaptations to existing security techniques and developing new methods to safeguard these advanced systems.

1.1.5 Security techniques for IoT

The security of IoT systems relies on a multi-layered approach, integrating various techniques to ensure data protection, integrity, and confidentiality. These techniques are essential in mitigating the myriad of threats IoT systems face, from data breaches to denial-of-service attacks. Key security techniques include encryption, authentication, access control, and intrusion detection.

Encryption (ENC) is the cornerstone of data security, converting information into a format that can only be read by those possessing the decryption key. In IoT, lightweight encryption algorithms such as Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC) are crucial due to the limited computational resources of many devices [15]. Ensuring that data remains encrypted during transmission and storage protects it from unauthorized access and tampering [16].

Authentication mechanisms verify the identity of devices and users accessing the network, preventing unauthorized entities from infiltrating the system. Techniques

such as Public Key Infrastructure (PKI), biometric verification, and multi-factor authentication enhance the security of IoT networks [17].

Access control systems (ACS) regulate which users or devices have permissions to access specific resources within the network. Implementing Role-Based Access Control (RBAC) or Attribute-Based Access Control (ABAC) ensures that sensitive data and critical functions are only accessible to authorized entities [18].

Intrusion Detection Systems (IDS) monitor network traffic and device behavior to identify potential security breaches in real-time. Anomaly detection algorithms and machine learning techniques can analyze patterns and detect unusual activities indicative of security threats.

1.2 State of contribution

After briefly introducing the principal aspects and for the rest of our thesis, it is apparent that we must address security, which enhances the healthcare domain. The question we need to answer is: How can data and communication security in the IoT environment be ensured? What security technique should be adopted, considering the IoT device's limitations? The answer to this question is what we aim to find out through our research. We will start by guaranteeing data transfer security among the elements of the IoT-Cloud environment. This will be accomplished by adapting the Homomorphic Encryption (HE), which allows us to perform operations on stored data without compromising its security. Simultaneously, we will verify that the suggested system, utilizing an effective encryption mechanism, can store data in a significantly reduced space compared to conventional encryption strategies. These are the main contributions of our research:

1.
   - Introducing a novel encryption method that combines data into a singular ciphertext by applying Multi-Key Embedded Encryption (MKEE). The efficacy of MKEE scales correlates with the amount of information contained in the ciphertext. To further reduce the size of the ciphertext in our method, a Modular Multiplicative Inverse (MMI) technique is suggested.
   - To validate the practicality of the proposed method, a case study was conducted on a diabetes dataset. By integrating MKEE and MMI, this study showed a data storage reduction of 94%.

2. Proposing a manuscript on a Deep-Learning (DL) model for detecting Tuberculosis (TB) using chest X-ray image classification. The model uses two

pre-trained deep-learning Convolutional neural networks (CNNs) and an ImageNet dataset, with a block attention module for spatial data. Experiments on four datasets showed excellent accuracy of 0.9966 and 0.9978 for training and validation sets, respectively.

1.3 Structure of the thesis

This thesis is divided into five chapters. Chapter 1 contains an introduction along with generalities in which we briefly mentioned some of the relevant notion. Chapter 2 is devoted to preliminaries where we presented the related mathematical and theoretical tools to this study. Chapter 3 represents our contribution which is demonstrated in the achieved results by suggesting a technique to solve the problem of maintaining data and communication safety while reducing the storage in the context of IoT. Chapter 4 introduces a deep learning model for early detection of Tuberculosis (TB), demonstrating how AI can improve the accuracy and speed of medical diagnoses. Chapter 4.4 is the final chapter in which we conclude our thesis along with a general statement of the achieved results.

1.4 Scientific contribution

Authored Paper- Published:

- Reducing the Encrypted Data Size: Healthcare with IoT-Cloud Computing Applications, Computer Systems Science and Engineering Journal, July 2024. [19]

  Authors: Romaissa Kebache, Abdelkader Laouid, Ahcene Bounceur, Mostefa Kara, Konstantinos Karampidis, Giorgos Papadourakis and Mohammad Hammoudeh.

Co-Authored Paper- Published:

- Semi-Decentralized Model for Drone Collaboration on Secure Measurement of Positions, ICFNDS '21: The 5th International Conference on Future Networks and Distributed Systems Dubai, UAE, December 2021. [20]

  Authors: Mostefa Kara, Abdelkader Laouid, Ahcène Bounceur, Mohammad Hammoudeh, Muath AlShaikh and Romaissa Kebache.

Conference Paper- Published:

- Tuberculosis Detection Using Chest X-Ray Image Classification by Deep Learning, International Conference on Future Networks and Distributed Systems (ICFNDS), American University in Dubai, December 2023.

  Authors: Romaissa Kebache, Sana Sahar Guia, Abdelkader Laouid, Mostefa Kara, Nassima Bouadem.

Chapter 2

Preliminaries

2.1 Introduction

By April 10th, 2020, The Coronavirus, known as COVID-19 the pandemic, left more than 1.5 million people infected causing the death of almost a million people worldwide and an unprecedented economic crisis World Health Organization (WHO). However, with the widespread development of technologies, the majority of the damage has been avoided by transferring essential businesses from the real world to the virtual world where everything is connected through the Internet [21]. Among the developing technologies that helped to cope with this radical change and facilitated this transfer is the Internet of Things (IoT).

2.1.1 IoT definition

IoT is defined as, a network of physical objects (things) that contain technology to communicate and sense or interact with their internal states (peer-to-peer) or the external environment [22]. Rapidly, the integration of IoT devices in the ecosystem made a huge impact on several sectors (Industry, Healthcare, Agriculture, Daily life functions). There were about 30,73 billion connected devices recently and 75,44 billion devices by 2025 [23]. As requested, the established communication using appropriate IoT protocols and its specific architecture (seen later in this section 2.1.2) is so high and typically exponential to the number of peers (About 21,5 billion active device connections by 2025) [24]. The growth of the Internet of Things had an exponential increase. Its rapid spread has affected the public [25]. "As shown in Figure 2.1, IoT applications vary depending on their field of use. Currently, there are five types of IoT applications; Consumer IoT, Commercial IoT, Industrial Internet of Things (IIoT), Infrastructure IoT, and Internet of Military Things (IoMT). For example, at the industrial level, IoT is mostly used to monitor the production line using its connected monitors. On the other hand, adopting IoT to real life involves many serious challenges [26]. The potential challenge is security which is considered the prior concern in many fields such as data breaches, side-channel attacks, and virus and data authentication [27, 28]. Implementing IoT systems, in reality, made them vulnerable to different new attacks (When the number of IoT devices increases, the number of attacks increases as well [29]) which drives researchers to think about adopting traditional security strategies and introducing new technologies to deal with data privacy and integrity issues (view in Section 2.1.4). The recent revolution of connected things (Objects) in many fields will improve the quality of services as well as reduce costs. In addition, connected systems and devices are most targeted by attacks [30], for that it is important to develop reliable solutions to ensure security. Nowadays, IoT system nodes can collaborate in gathering data at high efficiency yet at low cost [31] and achieving a consensus between nodes which is

able to improve the security of all devices in an IoT system. However, due to IoT device's limitations such as energy/power consumption, computation, memory, storage, and/or resource [32] several challenges obstructs their data consensus realization [33]. The established IoT standard architecture, and the IoT elements characteristics gave the flexibility to adapt it in any field as mentioned above but create a level of difficulty (such as an Enormous number of nodes and big IoT data, Decentralization and heterogeneity, and Unstable and unpredictable connections which are the major characteristics of IoT.

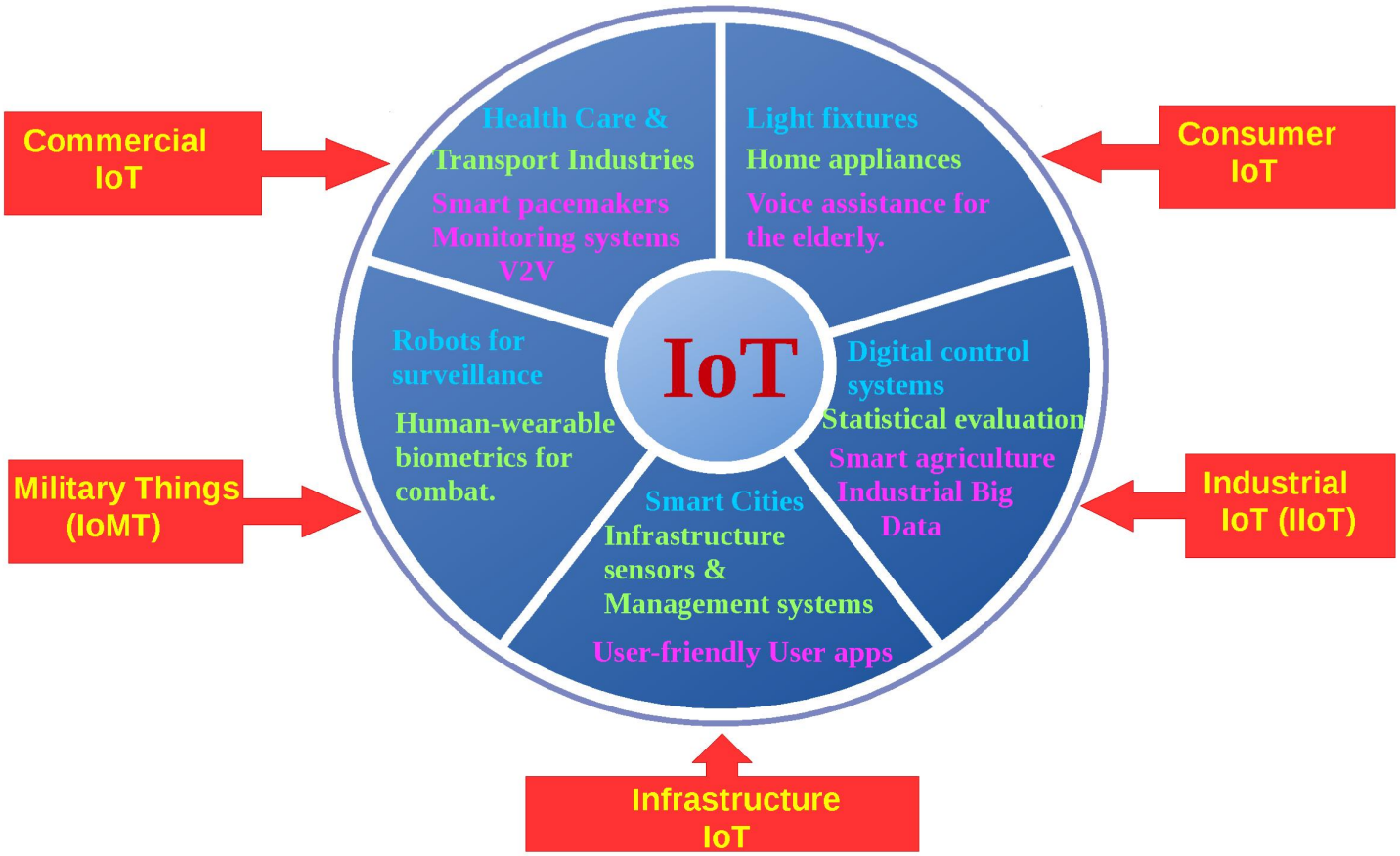

Figure 2.1: Major types of IoT applications

2.1.2 IoT architecture

As with any new technology, defining a standard architecture is crucial for implementation. The established IoT architecture provides flexibility for adaptation across various fields. However, its inherent challenges—such as a large number of nodes, massive data generation, decentralization, heterogeneity, and unpredictable connections—must be addressed. To manage these complexities, a typical IoT architecture consists of three main layers, or four layers in some applications: Perception (sensor layer), Networking, Service, and Interface Layer (see Figure 2.2).

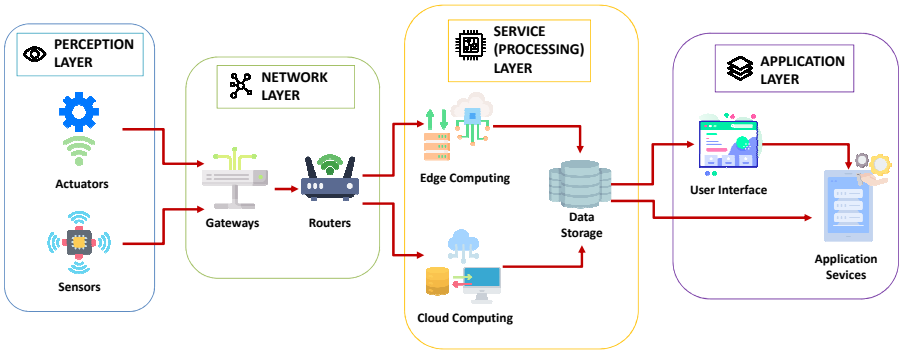

Figure 2.2: Main layers in IoT architecture

- The perception layer: This is a set of end devices, wireless sensors, and actuators identified using Radio-Frequency IDentification (RFID) technology that communicates via wireless Networks e.g.: Near Field Communications (NFC). Likely, this layer is the basic part that all IoT system relies on to connect the physical world and the digital world.
- The Networking layer: Is responsible for connecting other smart things, network devices, and servers.
- The Service layer: Creates and manages specific services to meet the requirements of IoT application.

- The Interface layer: Facilitates data use interactions with objects for specific applications.

2.1.3 IoT Security Aspects

As the Internet of Things (IoT) grows, protecting the devices and networks that connect to it is very important. IoT has many small, low-power devices, which can sometimes have weaknesses that bad people can use. To keep these devices safe, we need to use many methods like checking who is using them, keeping information private, and controlling who can access them. Table 2.1 summarises the main ways to secure IoT and why they are important for making a safe and strong IoT system.

Aspect	**Description**	**Importance**
Authentication	Ensures only authorized devices and users can access the IoT network.	Prevents unauthorized access and protects device integrity.
Encryption	Protects data by encoding it during transmission and storage to prevent unauthorized access.	Ensures confidentiality of sensitive data during transmission.
Secure Communication	Safeguards data exchanges between devices in the IoT network.	Maintains trust and integrity in data sharing across the network.
Access Control	Manages permissions for devices and users, ensuring proper security policies are enforced.	Limits exposure to security risks by restricting access.
Device Management	Monitors, updates, and controls IoT devices to maintain security and performance.	Keeps devices secure through regular updates and monitoring.
Data Privacy	Ensures that sensitive information is kept confidential and protected from unauthorized access.	Prevents data breaches and misuse of personal or sensitive information.
Threat Detection	Identifies and mitigates potential security threats within the IoT network.	Helps respond to security incidents in real-time to prevent damage.

Table 2.1: Key Aspects and Importance of IoT Security

2.1.4 Challenges in IoT Security

In the earlier section, we talked about how the Internet of Things (IoT) has changed many areas by making systems smarter and connecting them easily. But, the fast growth of IoT has also brought big security problems. Here, we will discuss the main recent issues in IoT security.

1. **Device Heterogeneity and Scalability Issues:** IoT systems include many types of devices, from basic sensors to advanced smart home setups. The wide range and large quantity of these devices make it hard to create a single security plan. Each device might have different hardware, software, and ways of communicating, which makes it tough to apply consistent security rules. Another issue is scalability; typical security methods often can't handle the huge number of devices in an IoT network.

2. **Resource Limitations:** Many IoT devices have limited processing power, memory, and battery life, which makes it difficult for them to run complex encryption or strong security measures. This means they are often easy targets for attacks because they can't handle the extra work that traditional security methods require.

3. **Lack of Standardization:** There are no common rules for how to keep IoT devices safe, so different devices and systems have different ways of protecting themselves. This makes it easier for bad people to find ways to attack, especially when devices from different companies don't have a single way to stay safe together.

4. **Privacy Issues:** IoT devices gather a lot of personal and important information. If this information isn't handled carefully or if there aren't enough safety measures, it can be stolen or misused. Privacy problems get worse because people often don't know how their data is being gathered, used, or stored.

5. **Physical Security Threats:** Unlike regular computers, many IoT devices are placed in areas where people can easily reach them. This makes them more likely to be tampered with, stolen, or physically damaged. Attackers can get close to these devices to steal data, put in harmful code, or use the device for illegal purposes.

6. **Lack of Software Updates and Patch Management:** Many IoT devices come with old or weak software, and their makers often don't provide regular updates or fixes. This lack of care leaves the devices open to known problems, making them easier to exploit.

7. **Network-Level Threats:** IoT devices are especially at risk from network attacks like Distributed Denial of Service (DDoS) and man-in-the-middle attacks. Because IoT devices are connected, a problem with one device could affect the whole network.

2.2 Mathematical Background

To provide a robust foundation for the theoretical and practical aspects of the security techniques discussed earlier. This section will cover the mathematical principles and methods necessary to understand and develop security algorithms and protocols in the context of IoT.

2.2.1 Related Mathematical Problems (Number theory)

Number theory is the study of integers and integer-valued functions. It forms the foundation of many cryptographic algorithms.

2.2.1.1 Prime Numbers and Factorization

Prime numbers are integers greater than 1 that have no positive divisors other than 1 and themselves. They play a crucial role in cryptography, especially in algorithms like Rivest–Shamir–Adleman (RSA).

The Fundamental Theorem of Arithmetic states that every integer greater than 1 can be uniquely factored into prime numbers such as Eq. 2.1:

$$n = p_1^{e_1} \cdot p_2^{e_2} \cdots p_k^{e_k} \tag{2.1}$$

where

- $p_1, p_2, ..., p_3$ are prime numbers
- $e_1, e_2, ..., e_k$ are positive integers (Exponents of the corresponding primes)
- The factorization is **unique** except for the order of the primes.

2.2.1.2 Modular Arithmetic

Modular arithmetic is the arithmetic of congruences. Two integers a and b are said to be congruent modulo n if n divides their difference, expresed as Eq. 2.2:

$$a \equiv b \pmod{n} \tag{2.2}$$

2.2.1.3 Discrete Logarithms

- **Generator**

  A *generator* g of a cyclic group G is an element such that every element of G can be expressed as g^k for some integer k. In other words, g generates the entire group by repeated application of the group operation. In the context of modular arithmetic, if $G = \mathbb{Z}_p^*$, then g is a generator if $g^k \bmod p$ yields all nonzero integers modulo p as k ranges from 1 to $p - 1$.

- **Discrete logarithm problem**

  The discrete logarithm problem (DLP) is a fundamental challenge in the field of cryptography. It involves finding the exponent x in the equation 2.3:

  $$g^x \equiv h \pmod{p} \tag{2.3}$$

  where g is a primitive root modulo p, and h is an element of the group generated by g.

  The security of many cryptographic systems, such as the Diffie-Hellman key exchange and ElGamal encryption, relies on the difficulty of solving the DLP.

2.2.2 Probability and Statistics

Probability and statistics are used in cryptographic analysis and the design of secure systems.

2.2.2.1 Probability Theory

Probability theory forms the mathematical foundation for statistics. It provides the tools to model and analyze random phenomena, which is crucial in evaluating the security of cryptographic algorithms.

For example, the probability P of an event A is defined as the following expression 2.4:

$$P(A) = \frac{\text{Number of favorable outcomes}}{\text{Total number of outcomes}} \tag{2.4}$$

Statistical measures such as mean, median, variance, and standard deviation help in analyzing the behavior of cryptographic systems under various conditions.

- **Random Variables and Distributions:** Cryptographic algorithms often rely on random variables and their distributions to ensure security properties like unpredictability and uniformity. For example, in the context of key generation, the keys must be uniformly distributed to avoid predictability.
- **Expected Value:** The expected value of a random variable X is a measure of the center of the distribution of the variable and is given by Eq 2.5 below:

$$E(X) = \sum_i x_i P(X = x_i) \tag{2.5}$$

  where

  - x_i are the possible outcomes of a random variable X
  - P are the probabilities of x_i means $P(X = x_i)$

- **Variance:** The variance of a random variable measures the spread of its values around the expected value and it helps in analyzing the or unpredictability of cryptographic outcomes. The variance is given by Eq 2.6.

$$\text{Var}(X) = E[(X - E(X))^2] = \sum_i P(X = x_i).(x_i - E[X])^2 \tag{2.6}$$

  where

  - X is a a random variable
  - $E[X]$ an expected value

2.2.2.2 Statistical Methods

Statistical methods are employed to test the randomness and security properties of cryptographic systems. Common methods include hypothesis testing, regression analysis, and analysis of variance (ANOVA).

Hypothesis testing, for instance, helps in determining if a cryptographic algorithm produces truly random outputs. The null hypothesis H_0 (e.g., "The outputs are random") is tested against the alternative hypothesis H_1 (e.g., "The outputs are not random").

- **Hypothesis Testing:** Hypothesis testing is a method of statistical inference used to decide whether a hypothesis about a parameter in a population is consistent with the data obtained from a sample. For instance, hypothesis testing can be used to determine if a given cryptographic algorithm produces output that is statistically indistinguishable from random noise.
- **Regression Analysis:** Regression analysis is a set of statistical processes for estimating the relationships among variables. It includes techniques for modeling and analyzing several variables when the focus is on the relationship between a dependent variable and one or more independent variables.
- **Markov Chains:** Markov chains model systems that transition from one state to another, which can be used in cryptanalysis to predict possible states. The transition from one state to another in a Markov chain is memoryless and only depends on the current state.

2.2.3 Information Theory

Let g be a generator of order q of a subgroup $G \subset Z$. By Lagrange's theorem, q is a divisor of $p-1$ because p is prime [34]. Given p, g, and $y \in G$, find x such that $y = g^x$. The so-called discrete logarithm hypothesis is therefore that it is difficult to compute x in a reasonable time if y is a random instance in Z. The numbers p and g are not necessarily chosen at random. The prime number p is selected to have certain properties that make it difficult to calculate the discrete logarithm. For example, p must be chosen such that $p-1$ has large prime divisors. The problem underlying discrete logarithm is the Calculatory Diffie-Hellman problem. Given g, $g^x \in G$ and $g^y \in G$, calculate $g^{x \times y}$

2.2.3.1 Entropy

Shannon entropy measures the uncertainty in a random variable X with possible values $\{x_1, x_2, \ldots, x_n\}$ and probability mass function $P(X)$ using Eq. 2.7:

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log P(x_i) \tag{2.7}$$

2.2.3.2 Mutual Information

Mutual information quantifies the amount of information obtained about one random variable through another which is calculated as Eq. 2.8:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{2.8}$$

2.2.4 Computational Complexity

Computational complexity theory is concerned with classifying computational problems according to their inherent difficulty and the resources needed to solve them.

2.2.4.1 Complexity Classes

Complexity classes categorize problems based on the resources required to solve them. Two of the most important complexity classes are P and NP:

- **Class P (Polynomial Time):** The set of decision problems that can be solved by a deterministic Turing machine in polynomial time ($O(n^k)$, where k is a constant).
- **Class NP (Nondeterministic Polynomial Time):** The set of decision problems for which a given solution can be verified by a deterministic Turing machine in polynomial time, even if finding the solution is difficult.
- **Class NP-Complete:** The hardest problems in *NP*. If any NP-complete problem can be solved in polynomial time, then all problems in *NP* can also be solved in polynomial time.

Other relevant complexity classes in cryptography include:

- **NP-Hard:** Problems that are at least as hard as the hardest problems in *NP*. They don't necessarily belong to *NP* (i.e., they might not be verifiable in polynomial time).
- **EXP (Exponential Time):** The set of problems solvable by a deterministic Turing machine in exponential time ($O(2^n)$ or worse).
- **PSPACE (Polynomial Space):** The set of problems solvable by a Turing machine using a polynomial amount of space (memory), regardless of the time required.

2.2.4.2 Algorithmic Efficiency

Algorithmic efficiency refers to the computational resources required by an algorithm to solve a problem. The efficiency of an algorithm is often expressed using Big-O notation, which characterizes the upper bound of an algorithm's runtime or space requirements in terms of input size n.

- **Time Complexity:** The amount of time an algorithm takes to complete as a function of the length of the input, which means Eq. 2.9

$$T(n) = O(f(n)) \tag{2.9}$$

  means that the growth of $T(n)$ is bounded above by $f(n)$ up to constant factors for large n.
- **Space Complexity:** The amount of memory an algorithm uses as a function of the length of the input.

2.3 Cryptography

Regarding personal information and every kind of private data, security is the primary concern. Nowadays, the Internet has become a necessity in our daily and professional lives since it is considered an essential communication tool for commerce, social interaction, and the transformation of personal information. As a result, the field of security is in continuous development. There are many aspects and applications to security. In this element, we will address one of the essential aspects of secure data and communications: cryptography.

2.3.1 Definitions

Cryptography is the science concerned with the design and analysis of techniques that ensure the confidentiality, integrity, and authenticity of information, even in the presence of malicious adversaries [35]. It provides mathematical foundations and algorithms, such as encryption, digital signatures, and hashing, to secure communication and data against unauthorized access and tampering. Etymologically speaking, the word Cryptography is derived from the Greek words "kryptos" and "graphein" which mean hidden and writing [36]. This aspect was used by Egyptians, as early as 1900 B. C., to hide the meaning from those who did not know the meaning [37]. Over time, Cryptography became an art and only recently have researchers established its primitive principles, functions, and objectives as shown in Table 2.2. As with any other field of study, Cryptography also was divided into sub-fields according to their usage such as cryptograms (using cryptography in puzzles), cryptanalytics (the science of code-breaking), and recently cryptocurrency [38].

Primitives	**Functions / objectives**	**Cryptanalysis: Attack Models**
*Authenticated encryption	*Privacy or confidentiality	*Cipher text only Attack
*Asymmetric algorithms	*Data Integrity	*Known Plain text Attack
*Constant time functions	*Entity authentication	*Chosen Plain text Attack
*Key derivation functions	*Message authentication	*Chosen Cipher text Attack
*Key wrapping	*Signature	
*Message authentication codes	*Authorization conveyance	
*Message digests (Hashing)	*Validation	
*Symmetric encryption	*Certification Endorsement	
*Symmetric Padding	*Time Stamping	
*Two-factor authentication	*Receipt acknowledgement	
	*Confirmation acknowledgement	
	*Anonymity	
	*Non- Repudiation	
	*Revocation	
	*Key exchange	
	Others:	
	*Forward Secrecy	
	*Perfect Security	
	*Deniable Authentication (Message Repudiation)	

Table 2.2: Cryptography Primitives, Functions/Objectives, and Cryptanalysis Attack Models

2.3.2 Cryptography classification

The emergence of modern cryptography came along with the rapid evolution of technologies in addition to the secret attempts for breaking codes which started in the 1930's by Polish cipher experts [39]. Their trial was on one of the most complicated cryptography machines at that time which is named Enigma. This machine was made by Germans and it was used in the war. However, just before the broke of war, the Polish cipher experts managed to break the codes of the

machine which introduced the main secret of cryptography development. The secret is that cryptography is not a perfect solution to security problems but a way to minimize them. From this point, Cryptography was divided into two main types which are classical and modern . Until WWII, classical cryptography was characterized by using pen and paper for ciphering by establishing preliminary cryptosystems and their terminologies [40]. However, with the appearance of computers, the ciphering task became easier since computers themselves do the work. The differences between classical and modern cryptography are represented in Table 2.3.

Classical Cryptography	**Modern Cryptography**
Works on Characters directly: letters and digits directly	Works on Binary Bits: (0 and 1 sequences)
Methods of Securing messages kept in secret and only communicators know about them	Methods and algorithms of Securing messages are public
All systems are required to be secret	Only the key needs to be secret
Less secure and some of it broken	Much more Secure

Table 2.3: Main characteristics of Classical & Modern Cryptography

The basic goal of cryptography is to ensure the security of communications across an insecure medium. For that, there are multiple methods and techniques that not only ensure the security of communications but also make cryptosystems non-vulnerable. Among these methods and techniques, there is encryption which is considered the most confusing term when defining cryptography. While cryptography is a field of study, encryption is a mathematical operation. It was defined as "a technique which ensures that data is visible only to the sender and recipient and no middle man can steal the data and snoop for information". In other words, it converts the information into secret code that hides the information's true meaning. Thus, cryptography is the science of encrypting and decrypting data. In computing, unencrypted data is also known as plain text, and encrypted data is called ciphertext. The formulas used to encode and decode messages are called encryption algorithms, or ciphers (see figure 2.3) .

2.3.2.1 Lightweight Cryptography

Lightweight cryptography refers to cryptographic algorithms designed to be efficient regarding computational resources, power consumption, and memory usage. These algorithms are crucial for securing devices with limited resources, such as those in the Internet of Things (IoT), wearable technology, and embedded systems.

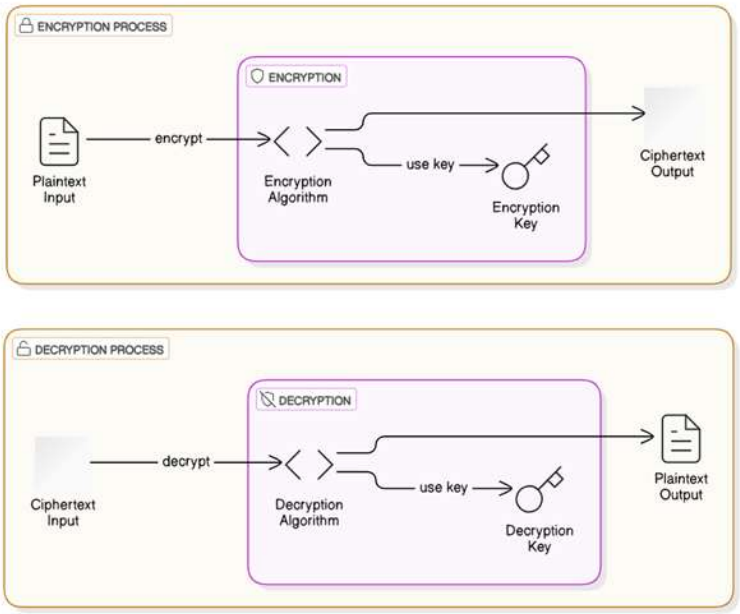

Figure 2.3: Encryption/Decryption Mechanism

Lightweight cryptographic algorithms are essential in various fields, particularly where devices are resource-constrained such as: IoT Devices, Medical Implants, Automotive Systems, and Smart Cities. The concept of lightweight cryptography emerged as a response to the increasing number of connected devices with constrained resources. Traditional cryptographic algorithms, while secure, often require substantial computational power and memory, making them unsuitable for small devices. In 2018, the National Institute of Standards and Technology (NIST) initiated a program to develop and standardize lightweight cryptographic algorithms, leading to the selection of Ascon as the standard in 2023 after a rigorous multi-year evaluation process [41, 42]

Key characteristics of lightweight cryptography include:

- Low Resource Consumption: Minimal processing power, memory, and energy usage.
- High Security: Strong resistance against various types of attacks, including side-channel attacks.
- Flexibility: Ability to perform efficiently across different hardware and software environments.

- Efficiency: Fast execution times, even on devices with very limited computational capabilities

So far, several notable lightweight cryptographic algorithms have been developed and evaluated explained in Table 2.4.

Algorithm	**Description**	**Characteristics**	**References**
Ascon	Family of algorithms for authenticated encryption and hashing.	High security, efficient, and robust performance.	Christoph Dobraunig et al. *Ascon v1.2: Lightweight Authenticated Encryption and Hashing* [43]
GIFT-COFB	Combines GIFT-128 block cipher with COFB mode for authenticated encryption.	Improved security over earlier ciphers, better performance.	Subhadeep Banik, Atul Luykx, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. *GIFT-COFB* [44]
Grain128-AEAD	Lightweight stream cipher with two shift registers and a nonlinear output function.	Low gate count, power consumption, and memory.	Martin Hell, Thomas Johansson, Alexander Maximov, and Willi Meier. *Grain-128AEAD v1.0* [45]
PHOTON-Beetle	Uses a sponge-based method for encryption and hashing, optimized for small-footprint devices.	Extremely small ROM and RAM usage, efficient for low-resource devices.	Zhenzhen Bao, Abhijit Chakraborti, and Yu Sasaki. *PHOTON-Beetle v1.0* [46]
ISAP	Lightweight block cipher focused on robustness against power analysis and fault attacks.	Uses Encrypt-then-MAC design, small code size, resistant to side-channel attacks.	Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. *ISAP v2.0* [47]

Table 2.4: Overview of key lightweight cryptographic algorithms, their descriptions, characteristics, and relevant references.

2.3.2.2 Public Key Cryptography

Public key cryptography (PKC), also known as asymmetric cryptography, is a cryptographic system that uses pairs of keys: public keys, which may be disseminated widely, and private keys, which are known only to the owner. This system allows for secure communication and data exchange over insecure channels without needing to share a private key in advance. The concept of public key cryptography was introduced in 1976 by Whitfield Diffie and Martin Hellman in their seminal paper "New Directions in Cryptography." This innovation solved the key distribution problem inherent in symmetric key cryptography by introducing the notion of a public key for encryption and a corresponding private key for decryption [48]. The RSA algorithm, developed by Rivest, Shamir, and Adleman in 1978, was one of the first practical implementations of public key cryptography [49]. Asymmetric key cryptography has essential characteristics which are:

- **Key Pair Usage:** Each user has a public-private key pair. The public key is used for encryption or signature verification, while the private key is used for decryption or signing.
- **Security:** The security of public key cryptography is based on mathematical problems that are easy to perform in one direction but difficult to reverse without specific information (e.g., factoring large prime numbers for RSA).
- **Authentication:** Public key cryptography provides a mechanism for verifying the identity of the sender through digital signatures.
- **Confidentiality:** It ensures that only the intended recipient, who possesses the private key, can decrypt the message.
- **Integrity:** Digital signatures ensure that the message has not been altered since it was signed.

The mechanism for encrypting a message using PKC has four main phases including:

1. ***Key Generation:***
    - Alice generates a pair of keys: a public key and a private key.
    - The public key is shared with anyone who wants to send a secure message to Alice.
    - The private key is kept secret by Alice.
2. ***Encryption:***

- Bob wants to send a secure message to Alice. He uses Alice's public key to encrypt the message.
- The encrypted message can only be decrypted by Alice's private key.

3. ***Transmission:***
    - The encrypted message is sent over the network. Eve, a potential eavesdropper, can intercept this message but cannot decrypt it because she does not have Alice's private key.
4. ***Decryption:***
    - Alice receives the encrypted message and uses her private key to decrypt it, retrieving the original message.

2.3.2.3 Symmetric Key Cryptography

Symmetric key cryptography (SKC), also known as secret key cryptography, uses the same key for both encryption and decryption of messages. This type of cryptography relies on the security of the key, which must be shared between the communicating parties through a secure channel. As a brief History, the origins of symmetric key cryptography date back to ancient times with techniques such as the Caesar cipher. In the modern era, symmetric encryption saw significant advancements with the development of block ciphers and stream ciphers. Key milestones include the Data Encryption Standard (DES) in the 1970s, the International Data Encryption Algorithm (IDEA) in the 1990s, and the AES in the 2000s, which remains widely used today [50]. SKC is characterized by:

- **Efficiency:** Symmetric key algorithms are generally faster and require less computational power compared to asymmetric algorithms, making them suitable for encrypting large amounts of data.
- **Security:** The security of symmetric encryption relies heavily on the secrecy of the key. If the key is compromised, the encryption can be easily broken.
- **Key Management:** One significant challenge is the secure exchange and management of keys. Each pair of communicating parties needs a unique key, which can become impractical in large networks.

Many old and recent cryptosystems are introduced, these are the most adopted for new symmetric key based-algorithms as mentioned in Table 2.5.

Cryptosystems	**Date of Creation**	**Description**	**Source**
AES (Advanced Encryption Standard)	2001	A widely adopted symmetric key algorithm that encrypts data in fixed block sizes (128, 192, or 256 bits) using a symmetric key of varying lengths.	Joan Daemen and Vincent Rijmen, *Rijndael for AES* [51]
DES (Data Encryption Standard)	1977	An older symmetric key algorithm that encrypts data in 64-bit blocks using a 56-bit key. It has largely been replaced by AES due to vulnerabilities.	National Bureau of Standards, *Data Encryption Standard (DES)* [52]
IDEA (International Data Encryption Algorithm)	1991	Known for its strength and efficiency, IDEA encrypts data in 64-bit blocks using a 128-bit key.	Xuejia Lai and James L. Massey, *A Proposal for a New Block Encryption Standard* [53]
Blowfish	1993	A block cipher that encrypts data in 64-bit blocks using variable-length keys from 32 bits to 448 bits. It's known for its speed and effectiveness in software implementations.	Bruce Schneier, *Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)* [54]

Table 2.5: Cryptosystems with Date of Creation and Source

2.3.2.4 Hash Functions

A hash function is a mathematical algorithm that takes an input (or "message") and returns a fixed-size string of bytes. The output, typically a "digest," is unique

to each unique input, and even a slight change in the input should produce a significantly different output. Hash functions are foundational in various fields such as cryptography, data integrity, and digital signatures. Hash functions have evolved significantly over the years, particularly with the rise of cryptography. Early hash functions like Message-Digest Algorithm (MD4), MD5, and Secure Hash Algorithm 1 (SHA-1) were widely used but later found vulnerable to various attacks. The vulnerabilities in these early hash functions led to developing more secure algorithms such as SHA-2 and SHA-3. The SHA-3 standard, selected through a public competition by NIST in 2012, marked a significant advancement in hash function design [55, 56].

Key characteristics of cryptographic hash functions include:

- **Deterministic:** The same input will always produce the same output.
- **Quick Computation:** Efficient to compute the hash value for any given input.
- **Preimage Resistance:** It should be infeasible to reverse-engineer the input from its hash value.
- **Small Changes in Input Produce Drastically Different Hashes:** Also known as the avalanche effect.
- **Collision Resistance:** It should be infeasible to find two different inputs that produce the same hash value.

Hash functions are crucial in various cryptographic applications, such as:

- **Digital Signatures:** Ensuring the authenticity and integrity of a message.
- **Password Hashing:** Securing passwords by storing hashes instead of plain text.
- **Data Integrity:** Verifying that data has not been altered.
- **Cryptographic Hashing:** Used in blockchain technologies for secure transaction recording.

Recent research has focused on enhancing the security and efficiency of hash functions. Lightweight hash functions have been developed for constrained environments like IoT, where computational resources are limited [56, 57]. Additionally, post-quantum cryptography has driven the development of new hash-based cryptographic systems that are resistant to quantum attacks, ensuring long-term security in the face of advancing computational capabilities.

2.3.2.5 Authentication Protocols

Authentication protocols are systematic procedures used to verify the identity of users, devices, or systems in a network. They ensure that the entities communicating are indeed who they claim to be. These protocols are crucial for maintaining security in various applications, particularly in environments like the Internet of Things (IoT), where numerous devices are interconnected [58]. The concept of authentication protocols dates back to the early days of computer networking and cryptography. Over the years, they have evolved significantly to address the growing complexities and security challenges of modern networks. Initial methods relied on simple password mechanisms, but with the advent of more sophisticated attacks, protocols have incorporated advanced cryptographic techniques.

In the 1980s, protocols like the Needham-Schroeder protocol laid the foundation for modern authentication mechanisms. The rise of the internet in the 1990s and 2000s necessitated more robust protocols like Kerberos and Secure Sockets Layer/Transport Layer Security (SSL/TLS), which use symmetric and asymmetric cryptography to enhance security. Today, with the proliferation of IoT devices, newer protocols are being developed to cater to the specific needs of low-power, resource-constrained environments. Effective authentication protocols exhibit several key characteristics, like:

- **Security:** They must provide strong protection against various attacks such as replay, man-in-the-middle, and brute force attacks.
- **Efficiency:** Especially in IoT environments, protocols need to be computationally efficient to conserve power and resources.
- **Scalability:** They should be capable of handling a large number of devices and users without significant degradation in performance.
- **Interoperability:** Protocols should work across different systems and platforms to ensure seamless communication and integration.
- **User-friendliness:** They should not overly burden users with complex procedures, ensuring ease of use and adoption.

Authentication protocols rely on cryptographic systems to secure communications. These can be broadly categorized into, Symmetric-Key Cryptosystems, Asymmetric-Key Cryptosystems, and Hybrid Systems. Where the last one aims

to combine symmetric and asymmetric cryptography to leverage the strengths of both. For instance, SSL/TLS uses asymmetric encryption to establish a session key, which is then used for symmetric encryption of data.

Recently, research has focused on developing lightweight authentication protocols tailored for IoT. These protocols aim to balance security and efficiency, addressing the unique constraints of IoT devices. For example, the CRO-Universal Security Authentication Protocol (CRUSAP) and Enhanced Security Rank Authentication Scheme (ESRAS) protocols have been proposed to meet these requirements, though they still face challenges in ensuring complete security against all potential attacks [59, 60].

2.3.2.6 Key Management Systems

Key Management Systems (KMS) are frameworks that handle the creation, distribution, storage, and maintenance of cryptographic keys throughout their lifecycle. They ensure the secure generation, use, and disposal of keys, essential for encryption and decryption processes in various security protocols. The evolution of KMS began with simple key storage solutions and manual key distribution. Over time, as networks and cryptographic practices grew more complex, automated systems emerged to manage keys more securely and efficiently. Early systems were often bespoke solutions for specific organizations. However, developing standards such as PKCS#11 and KMIP (Key Management Interoperability Protocol) helped streamline and unify key management practices across different platforms and applications. KMS utilizes the same cryptographic techniques mentioned in 2.3.2.5 to secure keys. The mentioned below are some of the essential specifics of KMS:

- **Security:** Ensures the confidentiality, integrity, and availability of keys. This involves protecting keys against unauthorized access, and alteration, and ensuring they are available when needed.
- **Scalability:** Must handle a growing number of keys and users efficiently without compromising performance.
- **Usability:** This should be user-friendly, allowing administrators to manage keys without extensive cryptographic knowledge.
- **Compliance:** Must adhere to industry standards and regulations to ensure data security and privacy.
- **Interoperability:** This should work seamlessly across different systems and applications, supporting a variety of cryptographic algorithms and protocols.

The news about KMS is that researchers have introduced innovative approaches to enhance security and efficiency. One notable advancement is the concept of Oblivious Key Management Systems (OKMS), which provide stronger security guarantees by hiding keys and object identifiers from the KMS itself. OKMS builds on Oblivious Pseudorandom Functions (OPRF), offering features like key verifiability and support for updatable encryption, enhancing forward and post-compromise security. In addition, lightweight key management protocols have been developed to address the unique constraints of IoT environments. These protocols aim to balance security with efficiency, catering to the limited computational and energy resources of IoT devices [61, 57].

2.3.2.7 Post-Quantum Cryptography

Post-quantum cryptography (PQC) refers to cryptographic algorithms designed to be secure against the potential capabilities of quantum computers. Unlike classical cryptographic methods, PQC aims to protect data from being easily decrypted by quantum algorithms, such as Shor's algorithm, which can efficiently solve problems that underpin the security of widely used cryptosystems like RSA and ECC. The concept of quantum computers was first proposed in the 1980s, but significant advancements in quantum computing have occurred over the last two decades. The development of quantum algorithms capable of breaking classical cryptographic systems prompted the need for PQC. In response, the National Institute of Standards and Technology (NIST) initiated a standardization process in 2016 to evaluate and select quantum-resistant cryptographic algorithms. This ongoing effort aims to establish standards that ensure long-term security in the quantum era [62]. Post-quantum cryptography (PQC) is characterized by:

- **Quantum Resistance:** PQC algorithms are designed to resist attacks from quantum computers. This involves ensuring that problems like integer factorization and discrete logarithms, which quantum computers can solve efficiently, are no longer the basis for security.
- **Classical Security:** While being resistant to quantum attacks, PQC algorithms must also remain secure against classical computational attacks.
- **Efficiency:** PQC algorithms need to be efficient in terms of computational resources and power consumption, making them feasible for widespread use across various platforms, including resource-constrained devices.
- **Interoperability:** These algorithms must be compatible with existing communication protocols and network infrastructures to facilitate a smooth transition from classical to quantum-resistant cryptographic systems.

Several classes of cryptosystems are being considered for PQC, including:

- **Lattice-based Cryptography:** Uses the hardness of lattice problems, which remain difficult for both classical and quantum computers, as the basis for security.
- **Hash-based Cryptography:** Relies on the security of hash functions, which are less affected by quantum computing capabilities.
- **Code-based Cryptography:** Based on error-correcting codes, offering strong security against quantum attacks.
- **Multivariate Quadratic Equations:** Uses the difficulty of solving systems of multivariate quadratic equations.
- **Supersingular Elliptic Curve Isogeny:** Involves the hardness of computing isogenies between supersingular elliptic curves.

NIST's ongoing standardization process is a significant milestone in the development of PQC. As of now, several candidate algorithms have been shortlisted, and rigorous evaluations are being conducted to assess their security, performance, and practicality [63]. Notable among the candidates are CRYSTALS-Kyber (for public-key encryption) and CRYSTALS-Dilithium (for digital signatures), both of which have shown promise in terms of security and efficiency.

2.4 Homomorphic cryptosystem

2.4.1 Definitions and motivation

In 1978, Homomorphic encryption was first suggested by Ronald Rivest, Leonard Adleman, and Michael Dertouzos [64]. It is a method that maintains data encryption during its processing and manipulation. This method does not need to reveal data values while implementing functions [65]. In other words, it provides a secured environment where operations can be performed on already encrypted data with the same results as on original data [66]. This is particularly useful for privacy-preserving computations on sensitive data. Thus, another concept of cryptosystem that was proposed is "privacy homomorphism" [67]. It provides direct computation on the encrypted data. This technique has two basic cryptosystems; multiplicatively homomorphic (evolved in the RSA Algorithm) and additively homomorphic (like in El Gamel Algorithm). However, partially homomorphic is when only one of the previously mentioned cryptosystems is used in

the encryption scheme (like in the DGHV cryptosystem) [68]. These approaches share the same term which is "error" or "noise".

Homomorphic encryption is a form of encryption that holds a unique property that is particularly valuable in IoT environments where data often needs to be processed by third-party services (e.g., cloud servers) without exposing sensitive information.

In IoT applications, devices continuously collect and transmit data to centralized servers for analysis. Homomorphic encryption can ensure that data remains confidential during this process. For instance, in healthcare IoT systems, sensitive patient data can be encrypted and then processed by cloud-based health analytics services without decrypting the data, thereby preserving privacy [69, 70]. Practical applications of homomorphic encryption in IoT also include secure aggregation of sensor data, privacy-preserving machine learning, and secure voting systems within smart cities. These applications demonstrate the potential of homomorphic encryption to provide robust security solutions while maintaining data utility and privacy [67]. To clarify the mechanism of the homomorphic encryption, an example using the Paillier Cryptosystem is scripted below.

The Paillier cryptosystem is a probabilistic asymmetric algorithm for public-key cryptography. It was invented by Pascal Paillier in 1999 and it contains three main phases including:

- **Key Generation**

  1. Choose two large prime numbers p and q such that $\gcd(pq, (p-1)(q-1)) = 1$.
  2. Compute $n = pq$ and $\lambda = \text{lcm}(p-1, q-1)$.
  3. Select a random integer g where $g \in \mathbb{Z}^*_{n^2}$ and ensure g satisfies the condition that $\gcd(L(g^\lambda \mod n^2), n) = 1$, where $L(u) = \frac{u-1}{n}$.

  The public key is (n, g) and the private key is λ. λ (lambda) is a part of the private key. It is calculated as the least common multiple (LCM) of $p-1$ and $q-1$, where p and q are the two large prime numbers chosen during key generation. The least common multiple of two numbers is the smallest number which is a multiple of both.

  In simpler terms, λ helps in the decryption process. It ensures that the computations work correctly when decrypting the encrypted message.

- **Encryption**
  To encrypt a message m (where $m \in \mathbb{Z}_n$), choose a random $r \in \mathbb{Z}^*_n$ and compute the ciphertext c as Eq. 2.10:

$$c = g^m \cdot r^n \mod n^2 \tag{2.10}$$

- **Decryption**
  To decrypt a ciphertext c, the plain text is m is calculated as mentioned in Eq. 2.11:

$$m = L(c^\lambda \mod n^2) \cdot \mu \mod n \tag{2.11}$$

  where $\mu = (L(g^\lambda \mod n^2))^{-1} \mod n$.

- **Additive Homomorphism Property** The Paillier cryptosystem is additively homomorphic. This means that given two ciphertexts c_1 and c_2, which are encryptions of m_1 and m_2 respectively, there exists a ciphertext c_3 such that Eq. 2.12:

$$c_3 = E(m_1 + m_2) = E(m_1) + E(m_2) \tag{2.12}$$

Proof.

1. Suppose $c_1 = g^{m_1} \cdot r_1^n \mod n^2$ and $c_2 = g^{m_2} \cdot r_2^n \mod n^2$ are encryptions of m_1 and m_2 respectively.
2. Consider the product of c_1 and c_2:

$$\begin{aligned} c_1 \cdot c_2 &= (g^{m_1} \cdot r_1^n) \cdot (g^{m_2} \cdot r_2^n) \mod n^2 \\ &= g^{m_1+m_2} \cdot (r_1 \cdot r_2)^n \mod n^2 \end{aligned}$$

3. Let $r_3 = r_1 \cdot r_2 \mod n$. Then,

$$c_1 \cdot c_2 = g^{m_1+m_2} \cdot r_3^n \mod n^2$$

4. Notice that $c_1 \cdot c_2$ is in the form $g^{m_1+m_2} \cdot r_3^n$, which is an encryption of $m_1 + m_2$ using randomness r_3.

Thus, multiplying the ciphertexts c_1 and c_2 yields a ciphertext that decrypts to $m_1 + m_2$, proving the additive homomorphism property of the Paillier cryptosystem. □

However, implementing homomorphic encryption in IoT is a field of challenges. The computational overhead associated with homomorphic encryption schemes is significant, making them resource-intensive. This is a critical consideration given the resource-constrained nature of many IoT devices. Research is ongoing to develop more efficient homomorphic encryption schemes that reduce computational complexity and enhance practicality for IoT applications [71, 72].

Recent advancements have led to the development of more practical homomorphic encryption schemes, such as the Brakerski-Gentry-Vaikuntanathan (BGV) scheme and the Cheon-Kim-Kim-Song (CKKS) scheme. These schemes offer improved efficiency and have shown potential for real-world applications, including secure IoT data processing [73, 74]. Additionally, the integration of homomorphic encryption with edge computing paradigms is being explored to offload intensive computations from IoT devices to more capable edge servers, thus balancing the load and enhancing overall system efficiency [75].

2.4.2 Types of homomorphic encryption

HE can be broken down into quite a few types based on operations done and the level of functionality offered. All these types are explained here, along with some of their mathematical backgrounds and examples [76, 74].

2.4.2.1 Partially homomorphic encryption

Partial Homomorphic Encryption schemes support either addition or multiplication, but not both. Examples like RSA Encryption and Paillier Encryption are explained below:

- ***RSA Encryption (Multiplicative Homomorphism):***

RSA supports multiplicative homomorphism. If $E(m) = m^e \mod n$ is the encryption of message m with public key (e, n), then $E(m_1 \cdot m_2) = (m_1 \cdot m_2)^e \mod n$.

Example: Given $m_1 = 5$ and $m_2 = 3$, and encryption key $e = 3, n = 33$, we get $E(5) = 5^3 \mod 33 = 26$ and $E(3) = 3^3 \mod 33 = 27$. The product of encrypted values $26 \cdot 27 \mod 33 = 6$ equals $(5 \cdot 3)^3 \mod 33$.

- ***Paillier Encryption (Additive Homomorphism):***

Paillier supports additive homomorphism. If $E(m) = g^m \cdot r^n \mod n^2$, then $E(m_1 + m_2) = E(m_1) \cdot E(m_2) \mod n^2$.

Example: Given $m_1 = 4$ and $m_2 = 5$, and encryption parameters $g = 2, n = 7$, the encryption $E(4) = 2^4 \cdot r^7 \mod 49$ and $E(5) = 2^5 \cdot r^7 \mod 49$. The sum of encrypted values $E(4 + 5) = 2^9 \cdot r^{2 \cdot 7} \mod 49$.

2.4.2.2 Somewhat Homomorphic Encryption (SWHE)

Somewhat Homomorphic Encryption (SWHE) schemes support both addition and multiplication but only for a limited number of operations before the noise becomes too large and the ciphertext needs refreshing. A prominent example is the BGV (Brakerski-Gentry-Vaikuntanathan) scheme.

- ***BGV Scheme:***

BGV uses lattice-based cryptography and allows both addition and multiplication on encrypted data. The plaintext space is usually a ring of polynomials $\mathbb{Z}_q[x]/(f(x))$.

Example: Given polynomials $p_1(x) = x+1$ and $p_2(x) = x^2+x$, their homomorphic sum and product are $E(p_1 + p_2)$ and $E(p_1 \cdot p_2)$.

2.4.2.3 Fully Homomorphic Encryption (FHE)

Fully Homomorphic Encryption (FHE) supports an unlimited number of both additive and multiplicative operations on ciphertexts, enabling arbitrary computations on encrypted data.

- ***Gentry's FHE Scheme:***

Introduced by Craig Gentry, FHE uses a bootstrapping technique to handle unlimited operations. The ciphertext space typically involves lattice-based structures, and the noise is managed via periodic "refresh" operations.

Example: For encrypted inputs $E(a)$ and $E(b)$, the scheme supports computing any function $f(a, b)$ homomorphically by repeatedly applying addition and multiplication and performing bootstrapping to manage noise.

- ***CKKS Scheme:***

CKKS (Cheon-Kim-Kim-Song) is designed for approximate arithmetic over real numbers, using a similar structure to BGV but optimized for real number computations.

Example: Encrypt real numbers r_1 and r_2, perform operations like $r_1 + r_2$ and $r_1 \cdot r_2$ directly on ciphertexts, yielding $E(r_1 + r_2)$ and $E(r_1 \cdot r_2)$.

2.5 IoT- Cloud environment

The domain that combines IoT (Internet of Things) and cloud computing is commonly referred to as Cloud IoT or IoT-Cloud. This integration leverages the strengths of both technologies: IoT devices' connectivity and data generation capabilities with the storage, processing power, and analytics capabilities of cloud computing.

IoT-Cloud refers to using cloud computing resources to manage, process, and store the vast amounts of data generated by IoT devices. This combination allows for scalable, flexible, and efficient handling of IoT data and enables advanced data analytics, machine learning, and real-time decision-making.

2.5.1 Characteristics of IoT-Cloud

- Scalability:

  Cloud platforms can dynamically scale resources to accommodate the fluctuating data volumes generated by IoT devices.

- ***Storage:***

  Cloud storage solutions provide vast amounts of space to store the large datasets generated by IoT devices over time.

- ***Processing Power:***

  Cloud computing offers powerful processing capabilities that can handle complex data analytics and machine learning tasks on IoT data.

- ***Accessibility:***

  Cloud services enable remote access to IoT data and applications from anywhere, facilitating seamless device management and data integration.

- ***Cost Efficiency:***

  By utilizing cloud resources, organizations can avoid the significant upfront costs associated with building and maintaining on-premises infrastructure.

2.5.2 IoT computing security requirements

The Internet of Things (IoT) represents a significant advancement in connectivity, allowing for the integration of various devices and systems through the Internet. However, this interconnectedness introduces substantial security challenges. IoT computing security requirements are essential for ensuring the confidentiality, integrity, and availability of data and services within IoT ecosystems.

1. **Confidentiality**

   Ensures that sensitive information is accessed only by authorized entities. This is crucial in IoT environments where data can include personal, financial, or health-related information.

   - Encryption: Encryption is the primary method for protecting data confidentiality in IoT. Symmetric encryption (e.g., AES) and asymmetric encryption (e.g., RSA) are widely used.
   - Access Control: Implementing robust access control mechanisms ensures that only authorized devices and users can access sensitive data. Techniques like role-based access control (RBAC) are common.

2. **Integrity**

   Ensures that data is not altered or tampered with during transmission or storage. This is vital for maintaining trust in the data exchanged between IoT devices.

   - Cryptographic Hash Functions: Hash functions like SHA-256 provide a way to verify the integrity of data by producing a fixed-size hash value from variable input data.
   - Digital Signatures: Digital signatures use asymmetric cryptography to ensure that data originates from a legitimate source and has not been altered.

3. **Availability**

   Ensures that IoT services and data are accessible when needed. This is critical for real-time applications such as healthcare monitoring and industrial automation.

- Redundancy and Failover Mechanisms: Implementing redundant systems and failover mechanisms can help maintain availability in the event of hardware or software failures.
- Denial of Service (DoS) Mitigation: Protecting IoT systems from DoS attacks involves techniques like rate limiting, IP blacklisting, and anomaly detection.

4. **Authentication**

   Ensures that the entities involved in communication are who they claim to be. This prevents unauthorized access to IoT devices and data.

   - Multi-Factor Authentication (MFA): Combining something the user knows (password), something the user has (token), and something the user is (biometric).
   - Lightweight Authentication Protocols: Protocols designed for resource-constrained IoT devices, such as the Constrained Application Protocol (CoAP).

5. **Authorization**

   Ensures that authenticated entities have permission to perform specific actions. This is critical in managing the access control policies for IoT devices.

   - OAuth 2.0: A widely used authorization framework that allows third-party services to exchange information without exposing passwords.
   - Attribute-Based Access Control (ABAC): ABAC uses attributes (e.g., user role, time of access) rather than fixed roles to grant access.

6. **Non-repudiation**

   Ensures that a party in communication cannot deny the authenticity of their signature on a document or a message they sent.

   - Digital Signatures: Using digital signatures ensures that the origin and integrity of the data can be verified, providing evidence in case of disputes.
   - Blockchain: Blockchain technology can provide immutable logs of transactions, ensuring non-repudiation.

2.5.3 Security threats in IoT environment

The Internet of Things (IoT) has revolutionized how devices interact and share data, providing unprecedented convenience and efficiency. However, the increasing

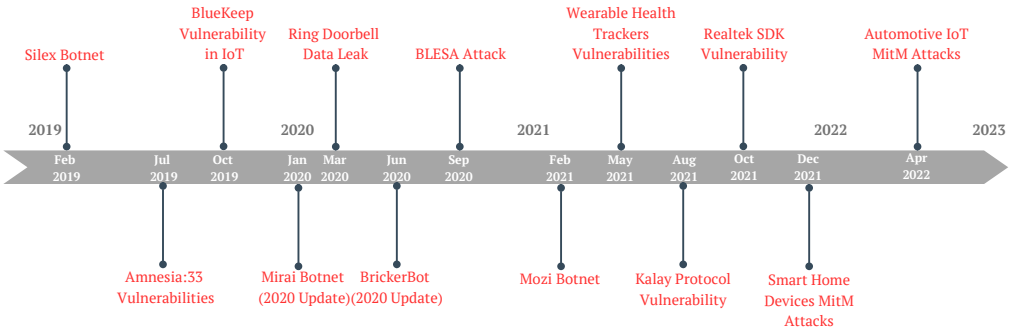

Figure 2.4: Timeline of IoT Security Attacks (2019-2023)

deployment of IoT devices has also expanded the attack surface for malicious actors. The Figure 2.4 provides a timeline of the primary security threats in the IoT environment.

2.6 IoT and Deep Learning Integration

2.6.1 Deep learning concept

Deep learning (DL) is a part of machine learning that works by copying the way the human brain's neural networks process information and make decisions. It uses layers of artificial neurons, called artificial neural networks (ANNs), to automatically find and learn patterns from large amounts of data. This layered structure allows deep learning models to handle complex tasks, such as recognizing images, understanding language, and interpreting speech, with great accuracy. Unlike traditional machine learning, deep learning models get better as they see more data, often needing less help from people to prepare the data for analysis.

2.6.2 Important Deep Learning Models and Techniques

2.6.2.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are commonly used for tasks in computer vision, such as classifying images, detecting objects, and analyzing videos. CNNs work by applying special filters to input data, typically images, to identify patterns and structures. They use layers like convolutional layers, pooling layers, and fully connected layers to effectively extract important features. A s an example of a modern model, **EfficientNet** from 2019 [77], adjust the width, depth, and resolution of the model to enhance performance in image classification tasks.

2.6.2.2 Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTMs)

RNNs, especially LSTMs, are strong models for data that comes in sequences, such as time series, speech, and text. RNNs can ”remember” past inputs through their connections that loop back, while LSTMs solve the problem of gradients becoming very small by using gates to manage memory over time. As example we have, LSTMs and their variation, **Bidirectional LSTMs**, have been used in tasks like creating text and translating languages, where understanding the context is very important.

Transformer Models Introduced in 2017, transformer models have greatly changed natural language processing (NLP). They use self-attention to manage long-distance connections in data, making them better for tasks like machine translation, text summarization, and question-answering. Models like **BERT** (Bidirectional Encoder Representations from Transformers) and **GPT-3** are now the best in NLP tasks.

2.6.2.3 Generative Adversarial Networks (GANs)

GANs are constructed of two parts, a generator and a discriminator, that work against each other. The generator makes fake data (like pictures), and the discriminator tries to tell if it’s real or fake. GANs such as; **StyleGAN2** (2020) [78] which generates realistic human faces and has been used in various creative and practical applications; are good at making images, changing styles, and creating high-quality data for training other systems.

2.6.2.4 Deep Reinforcement Learning

In deep reinforcement learning (DRL), an agent learns how to interact with its environment to achieve maximum cumulative reward. Deep learning techniques are applied to model the agent's decision-making process, enabling the learning of complex behaviors in dynamic environments. Recent models have surpassed human-level performance in games like Go and Starcraft like **AlphaGo** (2016) and **AlphaStar** (2019)

2.6.3 Convolutional Neural Network (CNN) Architecture

CNNs are built to automatically learn spatial patterns from input images by using operations such as convolutions, pooling, and activation functions. Figure 2.5 shows the structure of a Convolutional Neural Network (CNN), which is mainly used for classifying images.

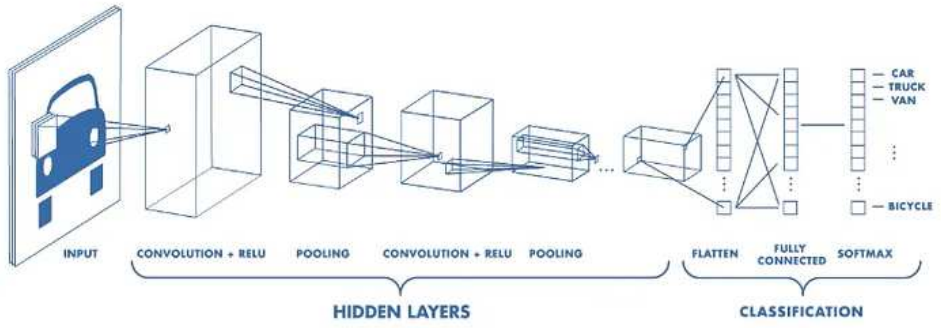

Figure 2.5: CNN Architecture.[1]

[1] Image source: https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html

1. **Input Layer** The input layer in this CNN architecture accepts an image, which is typically represented as a matrix of pixel values. In the example image, an illustration of a car image is used as the input. The size of the image might vary depending on the dataset, but the input format is a fixed-width and height matrix of pixel values.

2. **Convolution + ReLU Layers** The next step is the convolution operation, where filters (kernels) are applied over the input image to extract important features such as edges, textures, and other patterns. Each convolution operation generates an output feature map.

    - The ReLU (Rectified Linear Unit) activation function is applied to introduce non-linearity to the model, which allows the network to learn complex patterns. Without ReLU, the model would only be able to learn linear mappings.

    In the given image, this convolution step is applied multiple times, with each successive convolution layer working on the feature maps produced by the previous layer.

3. **Pooling Layer** After convolution, the next step involves pooling, specifically max-pooling, which reduces the spatial dimensions of the feature maps while retaining the most critical information. Pooling helps in decreasing computational load and mitigating the risk of overfitting.

    - The max-pooling operation extracts the most significant feature by selecting the maximum value from each patch of the feature map.

    This process of convolution followed by pooling repeats several times to extract increasingly complex features as the image passes through more layers.

4. **Fully Connected (FC) Layers** Following the series of convolution and pooling operations, the CNN architecture employs a fully connected layer, where the feature maps are "flattened" into a 1D vector. This vector is fed into the fully connected layers, which operate like a traditional neural network, making the final decisions based on the learned features.

    In the image, the output from the last convolution and pooling layers is fed into a fully connected layer, and these layers then predict which class the input image belongs to, whether it's a car, truck, van, or bicycle.

5. **Classification Layer** The final layer applies a softmax function to assign probabilities to the different classes. The class with the highest probability becomes the predicted label for the input image. In this case, the softmax layer outputs probabilities for each vehicle category.

2.6.4 VGG16 and VGG19 Architectures

A popular CNN architecture built on the basic convolutional layers is VGG16 and VGG19. These architectures are deeper versions of CNN, known for their simplicity and effectiveness in extracting hierarchical features from images. In addition, these models are commonly used as pre-trained models for transfer learning, where they are fine-tuned on new datasets, leading to excellent performance on various computer vision tasks.

1. ***VGG16 Architecture*** VGG16, as its name implies, consists of 16 layers (13 convolutional layers and 3 fully connected layers). It is characterized by:
   - **Small receptive filters (3x3)** in each convolution layer, which helps in capturing fine image details.
   - **A fixed structure** Each convolution layer uses 3x3 filters and is followed by a max-pooling layer, and the number of feature maps (channels) doubles after every few layers.
   - **Advantages:** Its deeper architecture improves the network's ability to capture more complex and abstract features, making it highly effective for large-scale image classification tasks.
2. ***VGG19 Architecture*** VGG19 is an extension of VGG16 and consists of 19 layers (16 convolutional layers and 3 fully connected layers). It follows the same design principles as VGG16 but adds more convolutional layers, making it even deeper:
   - **Advantages:** With more layers, VGG19 can capture more intricate details in the data. However, the increased depth also makes the model more computationally expensive and prone to overfitting without sufficient data.

   Both VGG16 and VGG19 architectures use fully connected layers followed by a softmax function to classify the image into different categories.
   - **Comparison:**
     - **VGG16:** Simpler, with fewer layers, lower computational cost, but still powerful.
     - **VGG19:**Deeper, with better accuracy but higher computational requirements.

The combination of the Internet of Things (IoT) and Deep Learning (DL) has created new opportunities in data processing, security, and automation. IoT systems involve a large network of connected devices that produce a huge amount of

data. But the complexity and size of this data require advanced techniques for analyzing, securing, and making decisions. Deep learning, which can automatically learn from large datasets and adapt to different tasks, is becoming a strong tool to enhance IoT applications. This part talks about how IoT and deep learning are currently working together [79], focusing on important topics like security, smart automation, and making data better.

2.6.5 The role of deep learning in IoT

Deep learning models are very good at tasks like recognizing patterns, anomaly detection, and classification, which are important for many IoT applications. These models, especially Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and autoencoders, are often used to analyze data from sensors, smart devices, and IoT systems. Recent studies show that deep learning has helped improve IoT in several ways, such as:

- **Data Analysis:** Deep learning has revolutionized the analysis of data in the Internet of Things (IoT) by automating the process of identifying important features and creating more accurate predictive models. This is especially important in fields like healthcare, where deep learning algorithms improve the continuous monitoring of patient data, offering real-time diagnostic insights. In IoT networks, deep learning can also enhance security. Systems for detecting intrusions (IDS) and identifying unusual activities, which are based on deep learning models, can spot unusual behavior patterns that might indicate cyber-attacks or system issues. A recent study developed an IDS that uses a combination of Long Short-Term Memory (LSTM) networks and autoencoders, significantly boosting the ability to detect attacks in IoT environments.
- **Optimization:** IoT devices usually have limited resources like battery power, memory, and processing speed. To handle this, deep learning methods, like using edge computing with smaller models, have been suggested to lower delays and make better use of resources. For example, MobileNet is a deep learning design made for edge devices, which helps create efficient IoT applications.

2.6.6 Security challenges in IoT and DL

The widespread use of IoT devices creates major security issues, mainly because of the variety of hardware, the large number of potential entry points for attacks,

and the limited processing power of many devices. Deep learning can help solve these problems by providing strong models that can identify and stop security threats as they happen [80, 81] The major fields that incountred challenges are : **Encryption and Data Security**, **Malware Detection**, **Authentication and Access Control**

2.6.7 Deep Learning in IoT Healthcare Systems

Integrating deep learning into IoT healthcare systems is one of the most impactful uses of this technology. By continuously gathering and analyzing vast amounts of data, deep learning can help diagnose diseases, forecast health issues, and manage patient information more effectively [82]

- **Medical Image Processing:** Deep learning has advanced in handling healthcare data from IoT devices, like medical images. One important use is applying CNNs to diagnose diseases using X-ray or MRI images. For example, a 2020 study used a CNN model combined with IoT devices to remotely detect pneumonia and tuberculosis from chest X-rays, achieving over 99% accuracy in real-time diagnosis.
- **Remote Patient Monitoring:** IoT devices with deep learning models are used to monitor patients' vital signs in real-time. These deep learning algorithms can spot unusual patterns in patient data and send alerts for early action. For example, a deep learning model in smart wearable devices accurately predicted heart events as they happened, giving healthcare providers time to act before a heart attack occurred.

2.6.8 Challenges and Future Directions

Although combining deep learning has a lot of possibilities, there are still some difficulties. These include:

- **Computational Resource Limitations:**
  Many IoT devices don't have enough power to run complex deep learning models. Edge computing and federated learning are good solutions, but more research is needed to make these models work better.
- **Data Privacy Concerns:**
  IoT networks produce a lot of sensitive information. Deep learning systems need to include privacy-protecting methods, like differential privacy and homomorphic encryption, in their designs.

- **Scalability:**
  As IoT networks keep expanding, the ability of deep learning solutions to handle large amounts is very important. Methods like model compression, pruning, and quantization are being studied to solve these problems.

2.7 Survey of Security Approaches for IoT

The rapid proliferation of IoT systems across critical domains—from healthcare to smart infrastructure—has exposed significant security vulnerabilities, necessitating robust countermeasures. This section synthesizes methodological trends in IoT security research, categorizing approaches into cryptographic solutions (e.g., homomorphic encryption for data privacy) and machine learning techniques (e.g., deep learning for threat detection). While cryptographic methods ensure data confidentiality and integrity in IoT-cloud environments, AI-driven approaches address dynamic threats through adaptive anomaly detection. By analyzing these methodologies, we highlight their strengths, limitations, and applicability to IoT's resource-constrained settings, paving the way for our proposed innovations in Sections 3 and 4

2.7.1 Homomorphic and Lightweight Cryptography

In the literature, several cryptographic techniques have been proposed. Playfair encryption [83] replaces each letter pair in the plaintext with another pair; for this, the Playfair scheme uses a square table (matrix) 5×5 built from a key, where each pair of letters gives the coordinates of a rectangle in the matrix. This creative method was not used very often since it could be easily deciphered by looking at which pair of letters appear most frequently in the ciphertext, assuming that they represent the most common pair of letters.

Hill's scheme [84] is used to encrypt the 26 letters using modulo 26. In ADFGVX systems [85], both techniques transposition and substitution have been mixed where the 26 letters of the alphabet, as well as the 10 digits, must be stored in a table of six boxes. Each letter in the plain text is replaced by a pair of letters corresponding to its row and column. The 3-rotor ENIGMA [86] is the most famous machine among rotor machines. This electric machine comprises an alphabetical keyboard, a light display, and three rotors. With each keystroke, the first rotor is rotated by one notch; at the end of a full turn, the second rotor is shifted by one, and so on. To define the encryption key, the rotors were positioned differently every time (FAC, for example). Data Encryption Standard Algorithm [51] (DES) is a block symmetric encryption algorithm that encrypts 64-bit words

given a 56-bit key (56-bit encryption + 8-bit parity used to verify the integrity of the key). For that, the plaintext message is split into 64-bit blocks. Each block bit is permuted according to the arrangement of the table. The 64-bit block is split into two 32-bit blocks denoted as G0 and D0, where G0 contains all the even bits of the initial message, and D0 contains all the odd bits. Rijndael also proposed Advanced Encryption Standard (AES) [87]. AES is a multi-turn block cipher similar to DES. However, AES uses larger, variable block and key sizes, such as 128, 196, and 256 bits.

The most essential problem of symmetric cryptography is the distribution of keys. If n people can communicate confidentially, then $n(n-1)/2$ keys are needed. The original idea of public-key cryptosystems was proposed by Rivest et al. [88], in which the fundamental principle is to use different encryption and decryption keys, unreconstructed from one another. Therefore, a public key is used for the encryption and a secret key for the decryption, where the public key plays the role of a padlock, and only the one that has the secret key can read the data. A major problem with this approach is that it is slower compared to asymmetric cryptography, which is nearly a thousand times faster. The RivestShamir-Adleman (RSA) encryption algorithm [89], based on the work of public-key cryptography of Diffie and Hellman, is a key exchange protocol, that allows two parties A and B, who are connected by an unsecured channel, to generate a secret cryptographic key. The key is difficult to find by an adversary intruding on the used channel.

In cryptography, there is also, the so-called homomorphic encryption [90] that allows us to perform operations on encrypted data without being decrypted [91]. Encryption is a crucial mechanism to maintain the confidentiality of any sensitive data. However, with the utilization of conventional encryption techniques, calculations on encrypted data are not applicable unless they are decrypted. Thus, the users must sacrifice their privacy to benefit from cloud services such as file storage, sharing, and collaboration. In addition, processes from servers, providers, and untrusted popular cloud operators may continue to physically identify their clients' data long after the client has terminated the relationship with the services. For that, this is a major privacy issue for customers. It would have been desirable if there was a mechanism that would not restrict the operations to be computed on the encrypted data while it was still encrypted; this can be achieved with homomorphism. The homomorphic properties can be shown in Eq. 2.13.

$$Dec(Enc(m_1)\Delta Enc(m_2)) = m_1\Delta m_2 \tag{2.13}$$

where m is the plaintext, $\forall\ m_1, m_2 \in m$ and Δ denotes addition or multiplication.

In [92], Gentry presented a fully homomorphic encryption technique using bootstrapping. Bootstrapping offers unlimited additive and multiplicative homomorphic operations. In 2019, Elgamal [93] proposed a new homomorphic encryption scheme for integer arithmetic using a variant based on the Chinese Remainder Theorem (CRT) secret sharing. RSA [[89] and Boneh [94] are the first feasible public key schemes as multiplicative cryptosystems, this property can be presented by Eq. 2.14.

$$Dec(Enc(m_1) \times Enc(m_2)) = m_1 \times m_2 \tag{2.14}$$

van Dijk et al. [95] proposed BGN (Boneh, Goh, and Nissim) scheme which supports an arbitrary number of additions and one multiplication represented by Eq. 2.15.

$$c = Enc(m) = g^m \times h^r \mod n \tag{2.15}$$

where m is the plaintext, g, h, and u are the generators with $(h = u^q)$, r is a random number and $n = p \times q$, where p and q are two prime numbers. Dasgupta et al. [96] proposed an FHE scheme that can be presented by Eq. 2.16.

$$c = Enc(m) = m + 2 \times r + p \times q \tag{2.16}$$

where $m \in \{0, 1\}$ and r is a random (r « p). The decryption operation is $m = Dec(c) = (c \mod p) \mod 2$. Doröz et al. [97] proposed an asymmetric homomorphic encryption scheme, making it fully homomorphic using a refresh operation. In [98], Doroz et al. proposed a leveled encryption based on a generalization of the open-source public-key cryptosystem NTRU (Number Theory Research Unit). Using blockchain technology [99, 100] and a hash function, the author of [101] encrypted images in an industrial IoT (IIoT) environment, encrypted data of more than 780,000 bits to encrypt a single bit. The puwhich needs solutions deeply optimized in terms of data processing and energy efficiency [102]. Most schemes generate a large amount of data or face network management and monitoring challenges [103] that may require public key size ranging from 70 Megabytes for the small setting and 2.3 Gigabytes for the large setting.

To improve data storage in cloud computing, Ahmad et al. [104] proposed a block-cipher-based anti-codify technique. the cipher text is generated using Deoxyribo Nucleic Acid (DNA) model. To raise the security level, the authors divide the original file into two separate blocks. In the field of securing healthcare data, Wei et al. [105] presented an image protection technique for healthcare applications to save patients' medical data transmitted in the Internet of Medical Things networks.

For more performance, this technique uses an enhanced 2D discrete chaotic map allowing dynamic substitution that is founded on an optimized nonlinear S-box.

While the encryption schemes mentioned earlier produce a substantial amount of encrypted data, this seeks to introduce an alternative technique. This method allows users to achieve reduced storage usage and minimized costs for the equivalent data employed in previous encryption schemes, all the while maintaining a high level of confidentiality and security.

2.7.2 Neural Networks in IoT Anomaly Detection

DL techniques have recently been regarded as the state of the art for image classification. Deep convolutional neural networks (DCNNs) have emerged as an appealing tool for image categorization among DL approaches. Many studies have recently employed CNNs for the automated identification of lung illnesses like pneumonia using CXR [106]. In the DL framework, concept transfer learning is utilized to identify TB using pre-trained models and their ensembles.

In PCXRNet [107], the authors proposed a new Attention-Based Convolutional Neural Network named PCXRNet, for pneumonia chest X-ray diagnosis. The network utilizes a new Condensed Attention Module (CDSE) and a Multi-Convolution Spatial Attention Module (MCSA) to extract useful information from chest X-ray images. PCXRNet is evaluated on three datasets: ChestXray2017, COVID-19 Radiography, and Tuberculosis Chest X-ray. The results demonstrate that PCXRNet achieved remarkable performance and outperformed other state-of-the-art models on multiple datasets. Therefore, they obtained an accuracy of 99.357% on the tuberculosis dataset. This was realized thanks to the proposed CDSE and MCSA modules, which are complementary, along with the optimal hyperparameters. In [108], the authors employed a modified version of the Unet model with Transfer Learning for segmentation and classification tasks. They removed one part at the end of both the encoder and decoder and extended the context part of the Unet model to contain some additional convolutional layers. Instead of simply concatenating between the features in the encoder part and the features in the corresponding decoder part, a bidirectional long-term convolutional memory has been placed. They used the tuberculosis dataset and achieved an accuracy of 98.6.

In [109], the authors proposed a deep neural network-based Tuberculosis diagnosis method which achieved a recall of 83.78% and precision of 67.55% for bacillus detection from microscopy photos of sputum. This technique takes a microscopy photo of sputum with a proper zoom level as input and returns locations of supposed Mycobacterium tuberculosis bacilli as output.

Paper [110] proposed a probable technique for tuberculosis detection utilizing DL which classifies CXR images into two classes; normal and abnormal. The authors used CNN architecture with 7 convolutional layers and 3 fully connected layers. They compared 99 the performance of three distinct optimizers. As a result, the Adam optimizer realized an overall accuracy of 94.73% and a validation accuracy of 82.09% which is considered best among the others. The obtained results are performed using Montgomery and Shenzhen datasets which are available in the public domain.

The authors of [111] proposed a tuberculosis detection model consisting of two subsystems; a data acquisition system and a recognition system. In the first system, a motorized microscopic phase is visualized to automate the acquisition of all FOVs. In this phase, movement is motorized and scanning patterns are designated by the user for a specimen examination. After the acquisition of all FOVs, data are handed to the second system. In this step, the transfer learning approach is implemented by customizing the Inception V3 DeepNet model. In this model, the fixed feature illustrations are taken from the top stack layer of Inception V3 DeepNet, and then, classified using SVM. The result showed an accuracy of 95.05%.

The authors of [112] trained the ensemble classifiers on images with similar features only. An ensemble needed a diversity of errors to perform well, which was achieved using either different classification techniques or feature sets. Tuberculosis detection using DL and Contrast-Enhanced Canny Edge Detected (CEED-Canny) X-ray images was proposed. The CEED-Canny was used to produce edge-detected images of the lung X-ray. Two types of features were generated; the first was extracted from the Enhanced X-ray images, while the second was from the Edge detected images. The suggested variation of features amplified the variety of errors of the base classifiers and enhanced TB detection. The presented ensemble technique gives an accuracy of 93.59%, sensitivity of 92.31%, and specificity of 94.87%.

In [113], authors proposed a TB diagnostics process that involved modifying CNN model structures, fine-tuning using an artificial bee colony algorithm, and implementing a linear average-based ensemble method. The model's performance on publicly available CXR datasets is validated, including the Shenzhen Hospital and National Institutes of Health datasets, by detecting lung abnormalities and distinguishing TB-related manifestations, with exceptional localization in CXRs. This study revealed that with continuous improvement steps, deep CNN models' performance improves, with structure modifications leading to the most significant increase in prediction accuracy, and fine-tuning. For this reason, the ensemble model outperforms other models in classification accuracy and robustness, even in disease localization tasks. It accurately provides attention maps to suspicious

lung areas, outperforming other state-of-the-art algorithms in both quantitative and qualitative results.

Likewise, the authors of [108] demonstrated the use of image pre-processing, data augmentation, image segmentation, and deep-learning classification techniques to reliably detect TB from chest X-ray images. For that, nine deep CNNs were trained and tested to classify TB and non-TB images (normal cases). The best-performing model, ChexNet, achieved 96.47% in terms of accuracy, precision, sensitivity, F1-score, and specificity. However, segmented lung images outperformed whole X-ray images, outperforming DenseNet201. The study suggests that the learning of CNN from the segmented lung regions can improve detection accuracy.

For more information on different employed methods for TB detection, an interesting review is given in [114]. It intends to explore the existing DL classifiers and assist researchers in developing a CAD system for the efficient diagnosis of TB. Therefore, different state-of-the-art DL techniques, that have been used to detect TB, have been explored and presented. Furthermore, the authors concluded that it is challenging to compare the methods with each other extensively because factors like the types of datasets used for evaluation, number of image samples used, evaluation metrics approach, and model parameters tunning have made the comparison complex.

Indeed, deep learning has multiple advantages, which has led to the emergence of a lot of research work in this area as proved in this section. However, few of them use a large-scale data set. For this reason, two articles have been worked on, that use the same large data set of 7000 images. Moreover, their results are among the most credible in the classification of tuberculosis.

2.8 Conclusion

The integration of Deep Learning (DL) and the Internet of Things (IoT) represents a transformative leap in the advancement of technology, poised to impact a multitude of industries from healthcare to security. This chapter has explored key areas essential to this integration, including the fundamentals of IoT, cryptographic techniques crucial for ensuring data security, the innovative approach of homomorphic encryption, and the synergies between IoT and DL. The study of IoT has shown its vast potential and the major challenges it faces, especially in managing and securing data. Cryptographic techniques, such as traditional encryption, are crucial for protecting data in IoT networks. But as IoT devices get more advanced, traditional cryptographic methods need to improve to handle the specific needs of these systems. Homomorphic encryption is a promising development because it allows calculations to be done on encrypted data, keeping

privacy intact while still enabling useful data analysis. This feature is especially important for IoT, where sensitive information needs to be kept safe from unauthorized access but still be available for analysis. The integration of DL with IoT has emerged as a crucial element in enhancing the functionality and efficiency of IoT systems. Recent advancements in DL have shown significant promise in improving data analysis, anomaly detection, and resource optimization within IoT networks. These advancements underline the crucial role of DL in addressing the complex challenges posed by IoT, especially in terms of enhancing security and optimizing resource utilization.

Chapter 3

Homomorphic encryption for IoT-cloud environment

In this chapter, we delve into the critical role of homomorphic encryption in securing IoT-cloud environments. The focus is on the innovative approaches to reducing encrypted data size, particularly in healthcare applications. The work "Reducing the Encrypted Data Size: Healthcare with IoT-Cloud Computing Applications, The Computer Systems Science and Engineering journal, 2024" serves as a key reference [19], exploring how homomorphic encryption can efficiently secure sensitive health data while maintaining system performance.

3.1 Introduction

Cloud computing is the technology that focuses on transferring, processing, computing, and storing data from a personal or local computer to a virtual space, which is a server device accessed through the Internet (Figure 3.1) [115]. This technology offers solutions to some critical issues like maintenance and development for both clients and companies since the client's efforts are solely devoted to utilizing its services. The cloud's infrastructure depends on advanced databases and provides significant resources for clients. The Virtual Cloud (VC) guarantees its accessibility (constant connection) to all devices anytime. With the increased development of technology available through the Internet, many private and public companies are seeking to deliver their applications by using cloud technology. After purchasing a certain space, the users can store their data on cloud servers' space, which can be accessible anytime and anywhere with an Internet connection.

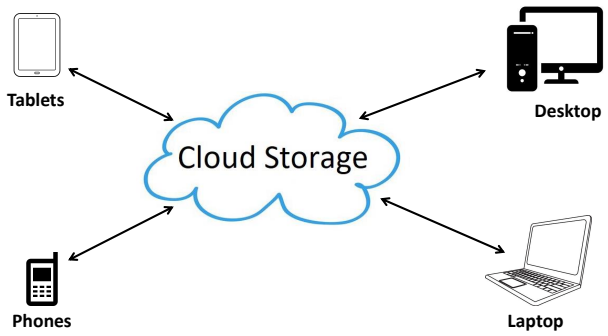

Figure 3.1: Cloud storage

Big data comprises digital data that are produced through the utilization of new technology for personal or professional purposes; it is primarily composed of compounded datasets derived from different sources. Big data could be considered a collection of diverse data, which is increasingly generated in greater volumes and at a higher rate. Consequently, traditional data processing software cannot handle

or store it locally. These large amounts of data can be provided to find solutions to issues that were previously considered to be unsolved.

The power and capacity of cloud computing lie in providing a tremendous advantage of storing, analyzing, and processing large amounts of data. The data gathered from intelligent hardware will be studied in multiple disciplines, examples include market basket analysis [116], web-analysis [117], healthcare [118], bioinformatics [119], and prediction [120]. Access to the cloud service saves considerable data collection costs; yet, data from multiple sources improves accuracy and efficiency in exploration results [121]. However, when cloud servers collect various data, they also collect sensitive client information; therefore, disclosing such data may harm users' privacy. Thus, data protection and security become crucial when storing them [122].

Cloud security is paramount to all data types, the majority of which are digital and should be only accessible by the owner [123, 124]. Information technology's advancements and increasing computational power raise many security concerns because of the volume, speed, and variety of data generated [125]. The collective use of cloud data has become increasingly important in the security domain, mainly in the encryption domain [126, 127, 128] . When discussing the process of exchanging or storing data, the time complexity and the size of the storage should be considered [129].

Healthcare data security is centered on safeguarding the information, computers, and networks utilized by healthcare providers and organizations. Viewing clinical data as a public asset leads to concerns about the protection and security of individual patient files. Ensuring the confidentiality of these records is crucial. The public's view of privacy regarding medical records is directly tied to their trust in the healthcare system at large and plays a significant role in debates about sharing health data.

Among the various security threats, false data injection attacks (FDIA) are very important. FDIA is concerned with the case where an attacker alters the data to conceal mistakes in calculating variables and values [130] . However, the influence of the FDIA on healthcare has been almost disregarded in recent years. A study conducted by IBM Security X-Force in 2023 report showed that reconnaissance activities, in which attackers search for security gaps and valuable data, are the major contributors to healthcare-related cyber incidents. Moreover, the recent increase in data security incidents in the healthcare sector has reaffirmed the necessity of preventive measures into account [131]. Hence, this demonstrates the importance of early-stage threat detection in healthcare. Confidential data might be intercepted or read by hackers in a data-sharing system such as a healthcare system in a cloud computing environment. Data cannot be secured by any other centralized control system since it cannot assume any third-party security infrastructure to protect this confidential data [132].

Data cybersecurity requires effective cryptographic systems to ensure data confidentiality, availability, and security. Nowadays, cloud computing is common in information technology, but storage and security issues have raised a challenge for users. Data encryption is an efficient solution, as most current encryption techniques support privacy and security [16]. Unlike most encryption techniques, some [133, 134] take the size of encrypted data that will be stored in the cloud seriously into consideration. Thus, as the size of the encrypted data increases, the higher it costs [135, 19].

In this work, an encryption method is presented that reduces the size of ciphertext to approximately 1/10 and 1/20 when compared against other schemes using the same parameters and size of data, which in return reduces the cost of storage while still preserving the confidentiality, privacy, and integrity of data that are stored in the cloud by private or public healthcare organizations. The proposed approach consists of grouping more than one information field into a single ciphertext by using a multiple secret keys technique. Furthermore, the proposal has been analyzed, and an equation that represents a reference to calculate the difference in the ciphertext size has been provided, reaching a rate of reduction equal to 88%. Moreover, to store data in the cloud at reduced costs, a case study of diabetes clinics has been performed. Another study on reducing the size using Modular Multiplicative Inverse was also conducted; this study is valid for almost any encryption technique with a storage space gain of up to 50%. Eventually, this will allow the owners of these clinics to store the data of their patients in the cloud securely and inexpensively.

The rest of this chapter is organized as follows. In Section 3.2, the proposed method is presented, while in Section 3.3, an analysis of the proposed technique is provided. Section 3.4 shows the conducted experiments and the overall performance of the proposed method. Section 3.4.1 discusses the size reduction by the inverse of a ciphertext. Section 3.5 presents a case study, and finally, in Section 3.6, a conclusion along with directions for future research work is given.

3.2 The Proposed Approach

In this contribution, a new concept of embedded encryption (EE) has been introduced. It is known that EE [136, 137] means to include an additional level of protection such as providing a physical layer of security or using multiple encryption algorithms. Indeed, these solutions are either financially expensive or not applicable in some environments, such as the Internet of Things. Typically, these solutions will eventually increase the volume of encrypted data and the processing time (the time required for the encryption and decryption processes). Embedding in the proposed technique is to hide a group of fields in a single field using several

small-sized keys. Fig. 3.2 shows the difference between the proposed encryption and other encryption schemes.

3.2.1 The Core of the Proposed Scheme

To simplify the understanding of our proposed Multi-Key Embedded Encryption (MKEE), it would be better to start by explaining the technique with a simple example shown in Fig. 3.3. In other systems, each entry is individually coded according to Eq. 3.1.

$$Enc(m) = function(m, N) \tag{3.1}$$

where m is the message to be encrypted and N is the public key. Most of the encryption methods [89, 94, 95] work with the modulo operation, which means that the size of the encrypted text will be in the range of the public key N. These techniques encrypt each message independently, i.e., if a record contains three fields (Fig. 3.3), each field will be considered as an independent input. Therefore, if there are three encrypted values, each of them must be in the range of the public key N. Thus, it is not possible to encrypt all fields (or inputs) in one ciphertext because these techniques are not linear; many of them are based on exponential functions (e.g., RSA) or they treat the bit level depending on multi-round process (e.g., AES). The proposed encryption scheme is a linear technique that contains multiplication and addition operations (see Algorithm 1).

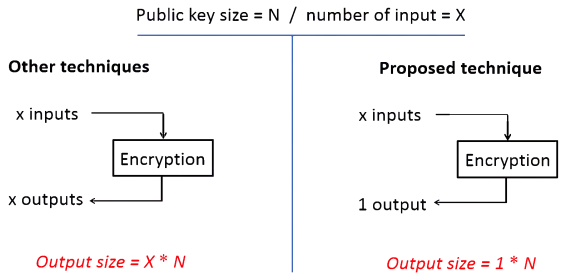

Figure 3.2: Input and output comparison

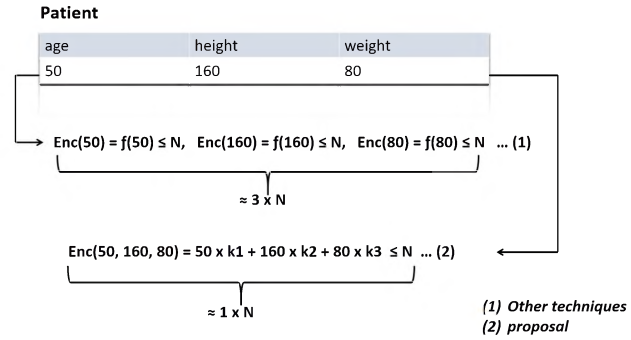

Figure 3.3: Example of encryption with three fields

Algorithm 1 Encryption algorithm

Require: public key N ▷ $N = p \times q$
Require: secret keys k_i
Require: plain texts m_i
Ensure: ciphertext c
1: generate a random number r
2: $c \leftarrow m_1 \times k_1 + m_2 \times k_2 + ... + m_i \times k_i + r \times p \mod N$ **return** c

Multiplication helps to mask the message value; the addition operation helps to separate hidden values so that can be retrieved during decryption. Eq. 3.2 shows the proposed symmetric encryption

$$Enc_{(sym,k_i)}(m) = (r \times p + \sum_{j=1}^{i}(m_j \times k_j)) \mod N \tag{3.2}$$

where $N = p \times q$ with p and q are two large prime numbers, r is a random number, $\forall\ j$, k_j is a secret key, and each m_j is a field or information such as First name, Last name, Age, etc. As shown in Eq. 3.2, the modulo operation is only performed once after hiding the fields and collecting them in a single ciphertext at the range of the public key N. In asymmetric representation, Enc() can be defined as follows:

$$Enc_{(asym,pk_i)}(m) = c = (\sum_{j=1}^{i}(m_j \times pk_j)) \mod N \tag{3.3}$$

where $pk_i = k_i + r_i \times p$. In fact, Eq. 3.2 $\leftrightarrow$ Eq. 3.3 because $Enc_{(asym,pk_i)}(m) = p \times \sum_{j=1}^{i}(m_j \times r_j) + \sum_{j=1}^{i}(m_j \times k_j)$. Assuming $R = \sum_{j=1}^{i}(m_j \times r_j)$, will get $Enc_{(asym,pk_i)}(m) = R \times p + \sum_{j=1}^{i}(m_j \times k_j)$. Eq. 3.4 shows the decryption process:

$$m_j = \frac{(c - (c \mod p))}{c} \tag{3.4}$$

The decryption process depends on the succession of the division's operations. m_j is the quotient obtained when we calculate $\frac{c}{k_j}$. After obtaining m_j, we calculate $c' = c - m_j \times k_j$; then, $m_{j-1} = \frac{c'}{k_{j-1}}$ and so on. Algorithm 2 shows how to go from a complete record which is represented by the ciphertext c to a set of information each represented by a plaintext m.

Algorithm 2 Decryption algorithm

Require: $record$, k_i, p
Ensure: $fields$
1: **function** DEC
2: $c \leftarrow c \mod p$
3: $fields \leftarrow ()$
4: **for** $j \leftarrow i$ to 1 **do**
5: $m_j \leftarrow \frac{c}{k_j}$
6: $c \leftarrow c - m_j \times k_j$
7: $inlist_{fields}(m_j)$
8: **end for**
9: **return** $fields$
10: **end function**

To ensure a mathematically correct decryption operation, three conditions must be met:

1. $m_i < k_i \; \forall \; i$
2. $\sum_{j=0}^{i-1} m_j \times k_j < k_j, \forall \; i, j$
3. $\sum_{j=0}^{i} m_j \times k_j < p, \forall \; i, j$

Proof. If $m_i > k_i$, then $(m_i \times k_i) \mod (k_i - 1) = \alpha$ where $\alpha < m_i$; therefore, m_i cannot be retrieved. Suppose that $c = m_1 \times k_1 + m_2 \times k_2$; when decoding, $c \mod k_2$ will be calculated first, then $c \mod k_1$. If $m_1 \times k_1 > k_2$, then $c \mod k_2 = \beta + m_2 \neq m_1 \times k_1 + m_2$, where $\beta < m_i$, the values m_1 and m_2 cannot be retrieved.

3.2.2 Homomorphic Addition Property

The proposed method is a holomorphic encryption that verifies the homomorphic addition property as shown in Eq. 3.5.

$$Enc(m_1) + Enc(m_2) = Enc(m_1 + m_2) \tag{3.5}$$

This homomorphic addition property is used in the sectors where privacy preservation is needed due to its importance in providing statistics on the user's data while respecting his confidentiality. For that, it satisfies the homomorphic addition as shown in Eq. 3.6.

$$\begin{aligned} Enc(m_1, m_2, ..m_i) &= m_1 \times k_1 + m_2 \times k_2 + ... + m_i \times k_i \\ Enc(m'_1, m'_2, ..m'_i) &= m'_1 \times k_1 + m'_2 \times k_2 + ... + m'_i \times k_i \\ Enc(m) + Enc(m') &= m_1 \times k_1 + m_2 \times k_2 + ... + m_i \times k_i \\ &\quad + m'_1 \times k_1 + m'2 \times k_2 + ... + m'i \times k_i \\ &= (m_1 + m'_1) \times k_1 + (m_2 + m'_2) \times k_2 + ... + (m_i + m'_i) \times k_i \\ &= Enc(m + m') \end{aligned} \tag{3.6}$$

Thanks to the proposed method, a user will be able to ask the cloud server to perform operations on the patient's data without access to its real value. For example, a user wants to calculate the average number of epilepsy times, $(\sum_{i=1}^{n} m_{1_i})/n$, where m_{1_i} are the number of epilepsy times. The cloud server will calculate $s = (\sum_{i=1}^{n} m_{1_i} \times k_1)/n$. Using the secret key k_1, the client will decrypt s as follows:

The average number of epilepsy times $= s \mod (k_1 - 1) = (\sum_{i=1}^{n} m_{1_i} \times k_1)/n$.

3.3 Security analysis

Ensuring robust security is a critical aspect of any encryption method, particularly in the context of IoT environments, where resource constraints and scalability are key concerns. This section provides an in-depth evaluation of the security properties of the proposed Multi-Key Embedded Encryption (MKEE) combined with the Modular Multiplicative Inverse (MMI) technique. We analyze its resistance to various attacks, its computational efficiency in securing IoT communications, and how it compares to existing encryption methods.

3.3.1 Comparison with Existing Encryption Methods

In this section, we compare the proposed Multi-Key Embedded Encryption (MKEE) combined with the Modular Multiplicative Inverse (MMI) technique against traditional encryption methods such as RSA, AES, and Elliptic Curve Cryptography (ECC). The comparison focuses on key aspects, as explained below.

3.3.1.1 Comparison Criteria

To evaluate the effectiveness of MKEE + MMI, we consider four key criteria that are essential for encryption in IoT environments. These criteria help assess the efficiency and security trade-offs of different encryption techniques.

- **Storage Efficiency:** The ability to minimize ciphertext size, which is crucial for IoT devices with limited storage.
- **Computational Cost:** The processing power required for encryption and decryption.
- **Security Strength:** The resistance against cryptographic attacks.
- **Suitability for IoT:** How well the method adapts to IoT constraints such as energy efficiency and real-time processing.

3.3.1.2 Comparison with Other Encryption Techniques

Table 3.1 presents a comparative analysis of MKEE + MMI with existing encryption methods, highlighting differences in storage efficiency, computational cost, security strength, and IoT suitability.

3.3.1.3 Discussion

The results in Table 3.1 demonstrate that MKEE + MMI significantly reduces ciphertext size by 94%, making it more storage-efficient compared to RSA and AES, which require larger ciphertexts. In terms of computational cost, MKEE + MMI requires fewer resources than RSA, making it suitable for low-power IoT devices, though ECC also provides efficient computation. Regarding security, while RSA and AES are widely used and offer robust encryption, MKEE + MMI enhances security by integrating modular inverse properties, making it more resistant to brute-force attacks. Finally, in terms of suitability for IoT environments, MKEE + MMI is specifically designed to balance security, efficiency, and storage constraints, making it a superior choice for IoT applications.

Encryption Method	Storage Efficiency	Computational Cost	Security Strength	Suitability for IoT
MKEE+MMI	High (94% ciphertext size reduction)	Low (Lightweight processing)	Strong (Modular inverse technique enhances security)	Excellent (Designed for IoT constraints)
RSA	Low	High (Key generation and encryption require large computations)	Strong	Moderate (Not optimized for IoT due to high computational needs)
AES	Moderate	Moderate (Fast encryption but requires more storage)	Strong	Good (Lightweight versions exist for IoT)
ECC	High	Low (Efficient key management)	Very Strong	Excellent (Used in constrained environments)

Table 3.1: Comparison of encryption methods based on key criteria.

3.3.2 Cryptographic Strength and Vulnerability Analysis

In cloud computing, confidentiality and preserving privacy problems are still challenging [138]. Generally, data stored in the cloud will be processed by servers. Such storage may be practical solely under a trusted cloud. Unfortunately, the data may be accessed, deleted, or manipulated by an untrusted provider. The security increases exponentially by increasing the number of files to be embedded in a single record. In this part, the proposal relies on the polynomial reconstruction problem robustness. In another part, our approach is based on the robustness of the number's factorization problem. In healthcare settings where data security and privacy are critical, we need to use a suitable number of fields besides utilizing secure prime numbers.

When the proposed method relies on the robustness of the number's factorization problem, i.e., the enemy has to factorize the public key $N = p \times q$ in order to get p or q, and eventually extract the secret key k. Thus, the secret key must be large, the trapdoor p also must be a safe and large prime number of form: $2 \times h + 1$ where h is a prime number. The original encryption is to multiply m by $k(m \times k)$. This technique is vulnerable to several attacks. By a couple $(Enc(m), m)$, the enemy can obtain the secret key k. Deterministic schemes are vulnerable to Chosen Plain text Attack (CPA) and they are not semantically secure. In CPA, the adversary

has the cipher text c and he tries to find the plaintext m. The adversary chooses m' and calculates $c' = Enc(m')$. Then, he decrypts $c \times c'$ to get $m \times m'$ and get m.

In the proposed method, $Enc(m) = c + c' = m \times k_1 + m' \times k_2$ where $k_1 < k_2$; if the enemy has m, he will multiply $Enc(m)$ by m^{-1}, he will get $k_1 + mm^{-1} \times m' \times k_2$; this does not give the enemy anything and he cannot get either k_1 or k_2 Even if the enemy has m and m', this will not help him in anything because the information encrypted by the multiplication operation is protected by the addition operation, and therefore neither of the two keys k_1 nor k_2 can be extracted. With the proposed method, there are i fields to encrypt in one value ($i > 2$). To get k_i, the enemy must have i^i messages with their corresponding encryption, which is not possible because each $m \times k$ is hidden inside. Therefore, the proposed technique is robust against chosen plaintext attacks and chosen ciphertext attacks.

We propose to hide a group of fields in a single field using several small-sized keys. Therefore, the factors that affect the proposed method are the size and nature of the data. On the one hand, the greater the size of the data, the greater the effectiveness because it follows the embedded encryption approach. On the other hand, the nature of the data may affect the effectiveness in terms of partitioning, because the input must be divided into fields so that each field is readable first, and second, the field length must be less than the key length.

MKEE + MMI provides a well-balanced encryption approach tailored to IoT environments by reducing storage needs while maintaining strong security and low computational overhead. While ECC remains a strong alternative due to its efficiency, MKEE + MMI offers superior storage reduction, making it a compelling choice for IoT applications.

3.4 Experimental and Results

In this Section, we will present the numerical results of MKEE and MMI techniques.

3.4.1 Size Reduction by Multi-Key Embedded Encryption (MKEE)

In the conducted experiments, the RSA cryptosystem was utilized for comparison since it is the most common and widely used encryption algorithm in the world. Two fields were initially encrypted, then three fields, etc., up to ten fields. A field

with 8 bits length and a public key with 1024 bits were used. In RSA, each field was encrypted separately. Therefore, to store two encrypted fields, a space equal to 2048 bits was needed; for three encrypted fields 1024 × 3 bits are needed, and eventually the required space is equal to 1024 × n, where n denotes the number of fields to be encrypted.

In the proposed method, the value of the first field is multiplied by the first key (size $(k1) = 1024$ bits), and the result is 1032 bits. That is, the size of the second key must be at least 1033 bits (3.1), therefore, the ciphertext size for both fields is 1033 bits. To encrypt the second value, it should be multiplied by the second key, which gives a value of 1041 bits, hence the size of the third key must be at least 1042 bits. Namely, encoding three fields gives us a total encrypted text of 1042 bits (compared to 1024 × 3 bits in RSA). To ensure encryption, the field size must be less than the first key size (k_1). Eqs. 3.7 and 3.8 formulate a general rule by which the total size of the encrypted text (the volume of storage needed) can be calculated.

$$size = k \times h...(others) \tag{3.7}$$

$$size = k + (h - 1) \times (l + 1)...(proposal) \tag{3.8}$$

where k is the secret key size, h denotes the number of fields, and l is the size of one field. The experiment in Fig. 3.4 shows the difference between the proposed encryption and RSA encryption schemes.

Fig. 3.5 shows how effective the proposed MKEE is in reducing the size of the digit and, therefore, the storage space. When encoding only two fields, the size's reduction rate is around 50%, because storing two RSA-encoded values requires 1024 + 1024 bits. On the other hand, in the proposed method, the ciphertext size is only 1033 bits (approximate text). The total size of the ciphertext in the proposed technique increases by 9 bits (nine is the field length plus one), unlike other techniques, where the total size increases by the addition of the size of the public key (1024, 2048, 3072, 4096, etc.). When encrypting 5 fields, the size reduction rate becomes 78% until reaching 88% when encrypting 10 fields, which is a very high ratio (0.5 vs. 5 kbits).

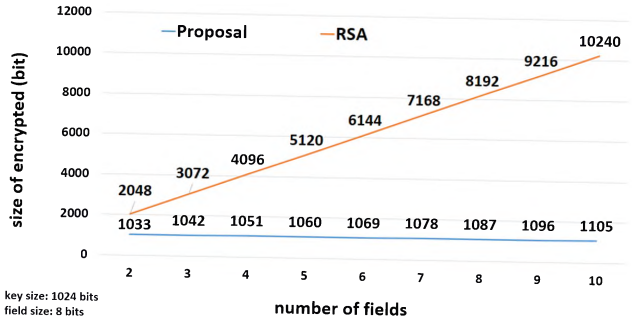

Figure 3.4: Size of ciphertext(s) in RSA and in the proposed encryption method

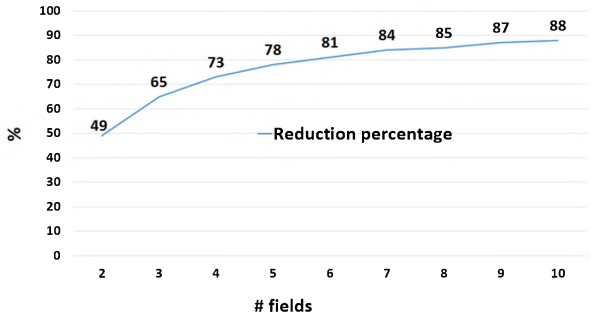

Figure 3.5: Percentage of size reduction by MKEE

On the other hand, we need ten multiplications and addition operations to encrypt 10 fields in MKEE. in contrast, ten exponential operations are needed in the RSA for the same number of fields. This makes our linear technique much faster in terms of execution time compared with the RSA cryptosystem.

3.4.2 Size Reduction by Multiplicative Inverse Method (MMI)

In addition to the experiments described in Section 3.4.1, more experiments that aim to gain storage space were conducted by using the Modular Multiplicative Inverse (MMI). Let x^{-1} be the MMI of x relative to a number p represented by $MMIp(x) = x^{-1}$, the MMI verifies the following Equation 3.9:

$$x \times x^{-1} \mod p = 1 \tag{3.9}$$

The idea was to replace the ciphertext c by its inverse c^{-1} if the size of c^{-1} is less than the size of c. A bit (0, 1) at each ciphertext is associated, for which this bit indicates whether this ciphertext is inverted or not (the value 0 means that the ciphertext is not inverted and the value 1 is the opposite). Algorithm 3 presents the utilized pseudocode.

Algorithm 3 Get Gain algorithm

```
1: s ← 0          ▷ It denotes the number of c^{-1} where size (c^{-1}) less than size (c)
2: ssc ← 0                    ▷ The sum of ciphertext sizes, sum(size(c_i))i = 1, 100
3: dsc ← 0        ▷ The sum of differences in sizes, i.e., sum(size(c) − size(c^{-1}))
                                          in case where size(c) > size(c^{-1})
4: for j ∈ (1, 100) do
5:     c ← Enc(m)                          ▷ The utilized encryption technique is
                                          Enc(m) = m1 + m2 × pk with m = m1 + m2
6:     ssc ← ssc + size(c)
7:     if size(MMI_n(c)) < size(c) then
8:         s ← s + 1
9:         dsc ← dsc + (size(MMI_n(c)) − size(c))
10:    end if
11: end for
12: gain ← (dsc/ssc) × 100
13: print(s, gain)
```

We can summarize this method as follows. The goal of MMI is to replace a ciphertext c by its multiplicative inverse c^{-1} when the size of c^{-1} is less than the size of c. It is an easy and lightweight operation to compute the MMI value. To further reduce the size of the ciphertext, this value can be computed according to the public modulo n or according to the private key p where $n = p \times q$. Tables 3.2 and 3.3 show these two uses of MMI (MMI_n and MMI_p), where the gain was 2% with MMI_n and 50% with MMI_p. Therefore, the trade-off associated with using MMI is the type of employing this technique.

Test	Size of n (digit)	♯ of c^{-1} where size $(c^{-1}) < size(c)$	Gain %
1	40	30	2
2	100	25	1
3	300	20	0.5

Table 3.2: Size reduction using MMI_n with 100 samples

Step 1
Experiments by calculating the MMI of c for the public key n have been performed, where $n = p \times q$. The results are shown in Table 3.2.

Test	Size of n (digit)	♯ of c^{-1} where size $(c^{-1}) < size(c)$	Gain %
1	40	100	50
2	100	100	50
3	300	100	50

Table 3.3: Size reduction using MMI_p with 100 samples

Step 2
Experiments by calculating the MMI of c for the private key p have been performed, where $n = p \times q$. The results are shown in Table 3.3.

By comparing Tables 3.2 and 3.3, it is evident that MMI_p achieved a very high result and it gave a space-saving of about 50% the size of the primitives k, p, q, and r (of which $pk = k + r \times p$ and $n = p \times q$). The calculation of the Modular Multiplicative Inverse for the private key p poses a risk of vulnerability with deterministic encryption. Storing c^{-1} (where $Enc_{pk}(m) = c$ and $c^{-1} = MMI_p(c)$) in an untrusted cloud, the adversary can extract sensitive information, i.e., he can get the private key p from an even (m, c^{-1}). In other words, if the adversary knows a single plaintext m and its corresponding ciphertext c^{-1}, he will calculate $MMI_n(c^{-1})$ to return to c.

Knowing that $MMI_n(c^{-1}) = c+\alpha \times p$ where α is a random number, the adversary will calculate $Enc_{pk}(m) = c$. The problem in deterministic encryption is that c in $(MMI_n(c^{-1}) = c + \alpha \times p)$ is the same in $(Enc_{pk}(m) = c)$, so the adversary will get $\alpha \times p$ and therefore p.

In probabilistic encryption, $Enc_{pk}(m) = c' \neq c$, i.e., $c-c' \neq 0$ where c' is calculated by the adversary using the public key pk with a random fragmentation of the plaintext m, and c is calculated by the user using his own random fragmentation of the plaintext m; this implies $c + \alpha \times p - c' \neq \alpha \times p$. Given the encryption example where the user's ciphertext is $c = m_1 \times pk + m_2 (m = m_1 + m_2$ with random fragmentation), the ciphertext generated by the adversary who knows m is $Enc_{pk}(m) = c' = m'_1 \times pk + m'_2$. So, $c-c' = m''_1 \times pk + m''_2 \neq 0$ (in more general terms, $\neq \alpha \times p$). Therefore, the adversary will not be able to obtain $\alpha \times p$. It should be noted here that there are techniques that can be called partially probabilistic. For instance, the technique $Enc(m) = c = m + r \times p$ where r is a random number at each message. In the general definition of probabilistic encryption, it is written as Eq. 3.10 :

$$Enc_1(m) = c \quad \text{and} \quad Enc_2(m) = c' \text{ with } c \neq c' \tag{3.10}$$

But this definition does not address a very important aspect in the field of security, which is calculating the difference between c and c', where $(c - c') \mod n$ can be equal to $\alpha \times p$, with α is a random number. That will give the adversary the possibility to get the private key p. On the other hand, there are encryption techniques that can be called fully probabilistic: $c = c'$ and $(c - c') \mod n \neq \alpha \times p$. For example, the encryption scheme where $Enc(m) = m_1 \times pk + m_2$ with $m = m_1 + m_2$ by random fragmentation is a fully probabilistic technique. Therefore, if there is a need to use the MMI_p, a fully probabilistic encryption technique should be utilized. By summarizing:

Partially Probabilistic Encryptiion (PpE): $c \neq c'$ and $(c - c') \mod n = \alpha \times p$
Fully Probabilistic Encryption (FpE): $c \neq c'$ and $(c - c') \mod n \neq \alpha \times p$

As a result, one can save 50% of the storage space in any FPE scheme if one uses the MMI. Indeed, the proposed MKEE is a deterministic schema with an asymmetric use, because $m_i \times pk_i \mod N$ does not change. MKEE is a partially probabilistic scheme with its symmetric use because $c = r \times p + \sum_{j=1}^{i}(m_j \times k_j)$ where r is a random number in each encrypted record. In order to take advantage of the MMI method, there is a need to transform MKEE into a fully probabilistic scheme. For this, the following new definition of MKEE is proposed as shown in Eq. 3.11:

$$c = (r + \sum_{j=1}^{i}(m_j \times k_j)) \mod N \tag{3.11}$$

where r is a small random number generated for each record, and $r < k_i$. Noting that $r = (c \mod p) \mod k_i$; so for decryption, $c - r$ should be computed, then the decryption function, that previously defined (Eq. 3.4), can be used. By this new formula, MKEE verifies the condition of FPE. Thus, combining MKEE and MMI gives us a reduction ratio for the size of the encrypted data estimated at 94%, i.e., it is approximately equal to 1/20 of the size obtained by using other encryption techniques.

Fig. 3.6 shows two curves, on the left, it illustrates the relationship between size and security. The security increases exponentially with the number of fields. On the right, the figure illustrates the relationship between size and encryption time. The encryption time does not increase significantly when the size is increased because the adopted technique is linear. We will increase one multiplication and one addition operation with each added field, which gives us 10 units of time per field.

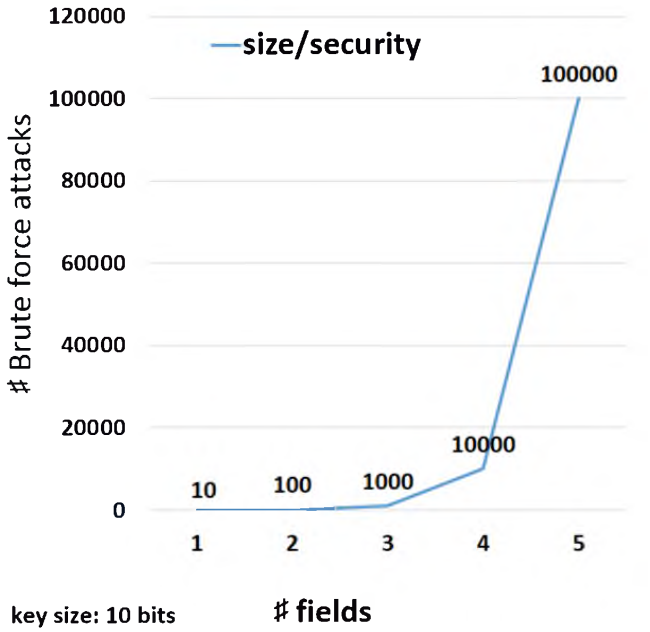

Figure 3.6: Comparison concerning memory size, security, and time

3.5 A Case Study: Diabetes Clinics

Nowadays, the importance of data in business and other sectors has increased. Data is the engine of customer relationships and mediation, the foundation of business strategy and any profitable project. Data management issues are a challenge for many organizations; however, the used data quality and the stored data quantity continue to be problems. The value of the data depends first on its quality and then on its size; therefore, these two factors are of great importance in the data world.

Correctly processed data delivers greater added value at all levels of the business. The strength of the encryption of a cryptography algorithm is generally linked to the size of its key: a large key results in more secure encryption. Advances in cryptographic analysis have largely influenced the increase in the key size used with algorithms. For example, in RSA, whenever the key size is doubled, the decryption operation requires six to seven times more processing power. Although it requires more computing power, the performance of computers, today, is sufficiently advanced to meet this demand.

Since January 2011, the Certification Authorities have tried to comply with the recommendations of the NIST (National Institute of Standards and Technology), by deciding that the key must be 2048 bits or more. Hence, some certification authorities have implemented the 2048-bit key length in their encryption systems.

Diabetes clinics offer health services that rely on a multidisciplinary team as well as pioneering research to ensure that the client receives the best care available. Comprehensive health assessments and treatments are coordinated by doctors, working with experienced nurses, social workers, and others. The clinics take care of diabetes problems and also provide psychological counseling to all groups. The clinics receive many patients and people looking for medical consultations every day (Fig. 3.7). Doctors and researchers benefit from this information by using statistics; therefore, these clinics should store all the data related to their patients and visitors. As a result, a huge amount of data is acquired, and this data is typically stored in the cloud. However, the more data stored in the cloud, the higher the cost the clinic pays.

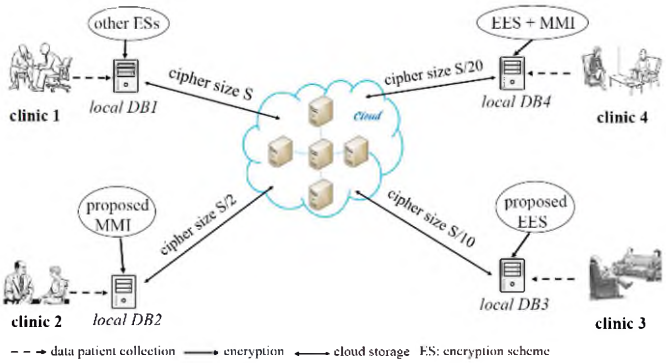

Figure 3.7: Proposed embedded encryption scheme (EES) and modular multiplicative inverse (MMI) in the use case

One of the clinics' goals, whether public or private, is to increase profits by reducing the share of expenses and payments. In this contribution, the proposed scheme reduces the size of the encrypted data to a rate of about 1/10 compared with other schemes, which is equivalent to 88% by grouping ten fields, that is ten values of information relating to one patient. For instance, this information could be the patient's name, address, age, weight, height, date of birth, date of examination, type of disease, and medication. Note here that this is just an example (the ten fields), and many clinics may use more information about a single patient. For example, when using twenty fields, according to the study conducted in Section 3.4, the percentage of size reduction will be equal to 87% ($1024 + 19 \times 81$ vs. 1024×20) since the key length is 1 kbits and the size of a field is 80 bits. With an estimation of the storage capacity needed for a clinic that receives 100 visitors

per day, and using the current size of the encryption key (2048 bits) in order to provide a high rate of security, an annual volume of data is estimated at:

$2kbits \times 20 \times 100 \times 365 = 1460Mbits$

Using the proposed method, the size will be:

$(2048 + 19 \times 81) \times 100 \times 365 = 130Mbits$

There is a big difference here in the capacity needed for storage and, of course, this will result in a big difference in the price needed for cloud-level storage. For example, iCloud, Amazon Cloud, and Google Drive provide an annual cost of between 15 dollars and 140 dollars per year for a capacity of 100 Gbits.

Finally, the research contributes in all fields to reduce the size of data, when we have a set of sensitive data units that can be read individually. Applications in healthcare include remote patient monitoring, individualized treatment strategies, and streamlined healthcare delivery. Also, for other types of sensitive data such as Personal and Private Customer, Employee, Financial, Business, and Operational data.

3.6 Conclusion

In today's digital landscape, data security, and efficient storage have become paramount, especially in scenarios where data is stored in paid cloud services. The cost implications of large-scale data storage have prompted innovative approaches to reduce both the size of encrypted data and the associated storage expenses. In this chapter, we presented an innovative encryption method designed to significantly reduce the size of the ciphered text. This size reduction is particularly advantageous for data storage in paid cloud services, as it results in substantial cost savings. The key strength of this technique lies in its robustness, achieved through the inherent correlation of the encrypted data, facilitated by the addition operation. The analysis of this approach yielded highly promising results, with an impressive 88% reduction in size when compared to traditional encryption techniques that encrypt each piece of information separately. This substantial reduction translates to a tenfold decrease in storage costs (1/10th of the original cost).

Furthermore, a study was conducted to explore the use of Modular Multiplicative Inverse for size reduction. The experiments conducted in this context showed a significant storage space gain of 50% when compared to certain other encryption techniques. To illustrate the practical application of our approach, we conducted

a case study on diabetes clinics, which routinely store data related to tens of thousands of visitors each year for future research and analysis.

This chapter has covered security in the IoT-Cloud Computing Applications at the storage level, the next chapter will deal with another aspect of security which is the security at the storage level; more specifically, data aggregation.

Chapter 4

Deep Learning for Enhance Healthcare Security

Tuberculosis (TB) continues to be a significant global health issue, particularly in areas where advanced diagnostic tools are not easily available. The growth of the Internet of Things (IoT) in healthcare means that smart devices can now gather and send patient data instantly. Combining deep learning models with IoT-based healthcare systems provides a new way to detect TB by analyzing chest X-ray images. These models improve diagnostic accuracy and allow for remote monitoring, while also protecting data through advanced security methods. Keeping sensitive medical data secure is very important, so secure IoT networks are necessary for the successful use of AI-driven diagnostics for TB and other illnesses. This chapter highlights a new deep-learning model for TB detection. The contribution is under the title: "Tuberculosis Detection Using Chest X-Ray Image Classification by Deep Learning, ICFNDS '23: Proceedings of the 7th International Conference on Future Networks and Distributed Systems, 2024"

4.1 Introduction

Tuberculosis (TB) is a deadly infectious disease initially caused by a bacterium called Mycobacterium tuberculosis. A healthy individual body can be infected with this disease, but it is known as latent tuberculosis and is not active until months or even years after the immune system becomes weak. Nevertheless, individuals with tired immune systems remain at greater risk of having this infection. When the bacteria enter the respiratory, they settle in the lungs. Since the immune system cannot protect the body because of its weakness, they start multiplying. In this case, the disease develops within days or weeks immediately after infection. For the average infected person, the bacteria multiply and attack the lungs, brain, spine, bones, kidneys, lymph nodes, and even the skin. Then it travels through the blood to the rest of the body [139]. This disease has many symptoms such as a cough that lasts more than three weeks, loss of appetite and unintentional weight loss, fever, chills, night sweats, high temperature, tiredness, and fatigue.

Artificial intelligence (AI), particularly DL (deep learning) algorithms, are attracting much attention because of their outstanding performance in image identification tasks. They can make a quantitative assessment of complex medical imaging automatically. Traditional machine learning algorithms and deep learning algorithms are the two types of AI technologies frequently employed in medical imaging. Combining AI and security including privacy [140, 141] and confidentiality [142, 143] in innovative solutions can give more efficient results.

DL has proved to be a very powerful tool because of its ability to handle large amounts of data. It has rapidly become the chosen technique in the field of

Radiology. Moreover, one of the most popular deep neural networks is Convolutional Neural Networks (CNNs). Obviously, the development of these networks has played a vital role in feature extraction for TB detection, in addition to the classification of Chest X-ray images as normal or abnormal. For that, medical researchers have begun employ DL in their work to detect diseases automatically.

Compared to traditional Computer Vision (CV) techniques, DL allows CV users to obtain more significant accuracy in studies such as image classification, semantic segmentation, entity detection, Simultaneous Localization, and Mapping. Since the neural networks utilized in DL are trained instead of programmed, applications utilizing this strategy often need less expert analysis and fine-tuning, in addition to exploiting the huge amount of data obtained in today's systems. DL also provides excellent flexibility because CNN models and frameworks can be re-trained utilizing a custom dataset for any use case, contrary to CV strategies, which tend to be more domain-specific.

The remainder of this chapter is organized as follows: In Section 4.1 we introduce the domain and relevant notions. . In Section 4.2, we will explain our proposed model before presenting our experiment result in Section 4.3. Finally, we conclude our work in Section 4.4.

4.2 The Proposed Model

Many different architectures are built in a specific order to be very effective in classifying images. Some of the mechanisms also help models extract spatial information from features, to make the model look in the right places during the classification process. This section discusses the used method and talks in detail about the basic parts of the model architecture.

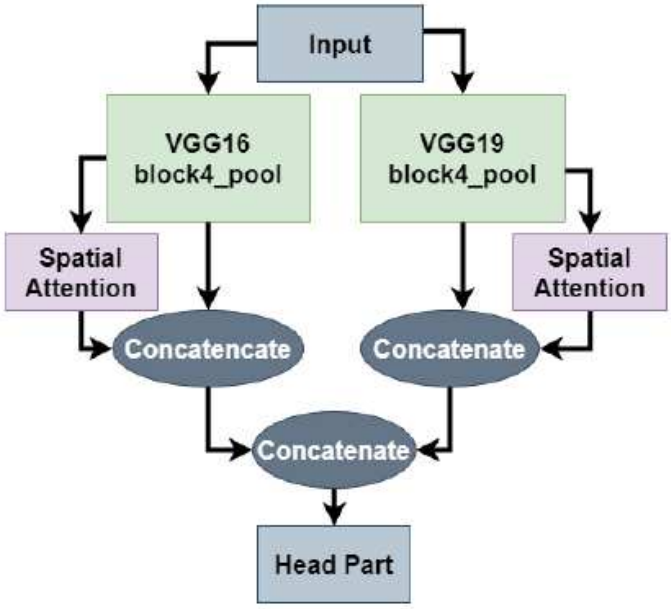

Figure 4.1: A component diagram of the proposed architecture for DL image classification.

An architecture that is inspired by the paper of [144] is created with modifications as follows:

- First, a model is created combined by two popular pre-trained models VGG16 and VGG19 on the Imagenet dataset as a backbone part.
- Then, the model's architecture (Figure 4.1) was augmented with some Attention Blocks to be able to learn spatial information.
- Finally, a dense classifier was added with a large number of units and a dropout technique to avoid the overfitting problem.

VGG (Visual Geometry Group) is a CNN model for large-scale image recognition that was proposed in [145]. Using the Imagenet dataset, this model achieved an accuracy equal to 92.7%.

The first part of VGG16 for feature extraction consists of 5 blocks, where each block contains a specific number of convolutional layers followed by one max-pooling layer in order to extract good feature maps from the input images. In the

last block, the model contains a set of fully connected layers responsible for the classification process [146].

VGG19 is a version that is very similar to VGG16 with changes in the number of layers (3 additional layers) which allows the extraction of features a bit deeper and makes it treat some problems more effectively.

To get the spatial attention map, the next steps are followed. First, the max pooling and avg-pooling are applied to the original features [147]. As a second step, the outputs of the first step are concatenated. Finally, a convolution module is applied to the output of the second step with specific arguments; (7×7) kernel and sigmoid activation function (Equation 4.1).

$$\begin{aligned} M_s(F) &= \sigma(f^{7*7}([AvgPool(F); MaxPool(F)])) \\ &= \sigma(f^{7*7}({F_{avs}}^s; {F_{max}}^s)) \end{aligned} \tag{4.1}$$

Equation 4.1 explains more about the process of extracting the spatial attention map, where the following symbols (σ,$f^{7\times7}$,F) represent respectively (sigmoid activation function, convolution layer with kernel of size 7, the input features). To get channel attention, as spatial attention, some simple steps are applied. In the first step, both max pooling and avg pooling are performed on the features. Then the second step, in which the two outputs of the first step are passed one by one to the multi-layer perceptron which has only one hidden layer from which the channel attention maps are obtained as a vector. After completing the second step, the channel vector maps are all merged together by element wise summation. Finally, a channel attention representation for the features entered in the first step is obtained.

$$\begin{aligned} M_c(F) &= \sigma(MLP([AvgPool(F) + MLP(MaxPool(F))])) \\ &= \sigma(W1(W0({F_{avs}}^c)) + W1(W0({F_{max}}^c))) \end{aligned} \tag{4.2}$$

In Equation 4.2, the ambiguous variables are explained as follows:W_0, W_1 are the multi-layer perceptron weights and σ is the sigmoid activation function.

The concatenate step takes a list of tensors as input, all of the same shapes except for the concatenation axis, and returns a single tensor that is the concatenation of all inputs. The head part is the part responsible for classification in the network. Therefore, a combination of layers that are able to classify the images well must be chosen and common problems such as over-fitting and under -fitting must be avoided. The head part contains four layers with certain arguments; Flatten Layer, Dropout Layer with 0.5 rates, Dense Layer with 256 units and ReLU activation function, Dense Layer with 2 units and sigmoid activation function.

4.3 Experiment Result

The used dataset contains four popular accessible datasets which are NLM dataset, Belarus dataset, NIAID TB dataset, and RSNACXR dataset. The total number of images that are obtained from the four datasets is 7000 images, which are divided into three groups as shown in Table 4.1.

	Tuberculosis	Normal	Total	Percentage
Test	700	700	1400	20%
Train	2100	2100	4200	60%
Val	700	700	1400	20%
Total	3500	3500	7000	100%

Table 4.1: Divided groups images

One of the common metrics in evaluating the performance of binary classification models is the ROC curve. To get the ROC curve, the model classification threshold is changed each time, and the values of the false positive rate and true positive rate are represented as a point in the space. As a result, the curve shows the model's ability to accurately classify and detect disease despite different thresholds.

The proposed method is applied with a set of configurations that help the model to be trained properly. These configurations are as follows: Adam Optimizer, Learning rate equal to 1e-3, Batch of size 32, Binary Cross-entropy Loss Function, Cross Validation with 4 folds, and Categorical accuracy metric to stop training in specific thresholds and save weights.

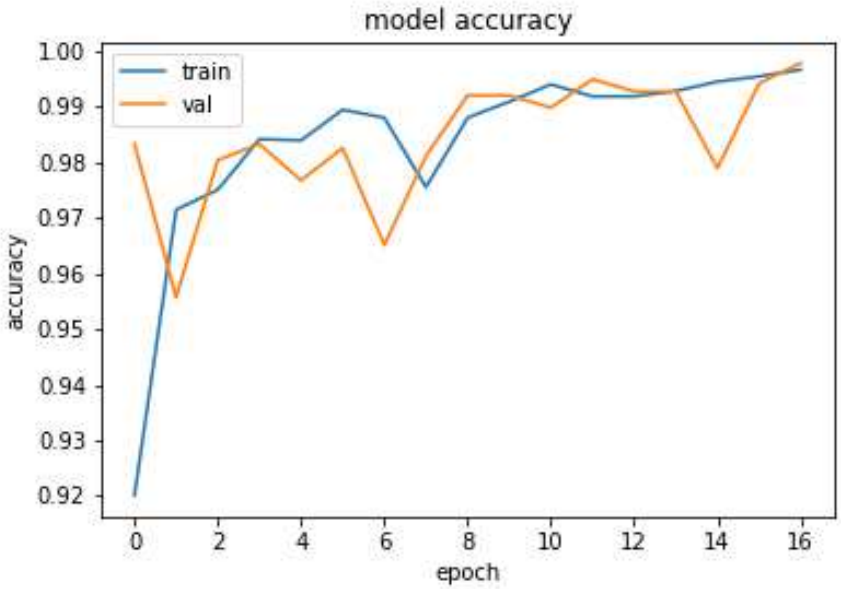

Figure 4.2: The model accuracy curve for both train and validation sets.

As is shown in Figure 4.2, the accuracy curve of both the training and validation sets increases until it reaches an excellent accuracy of 0.9966 and 0.9978 respectively. A callback class is created with the purpose of stopping training while reaching certain values of the accuracy and loss function of the validation set Code Listing. The architecture of the model is created, compiled, trained, and eventually evaluated. These steps are repeated several times on different sets of data. After training the model for several epochs, good results were obtained (see Figure 4.2).

4.4 Conclusion

CNNs have frequently outperformed other traditional recognition algorithms in image-based classification and recognition issues as deep learning develops. Therefore, they are the preferred choice for complex medical problem-solving to automatically extract valuable information from the inherent qualities of data, due to their exceptional capacity. In this chapter, a deep-learning model was proposed for Tuberculosis detection using chest X-ray image classification. Moreover, an

architecture for the proposed model was created that combined two CNNs models which are VGG16 and VGG19, in addition to the attention mechanism. The major aim was to build a model with a robust architecture in order to effectively solve the problem of identifying tuberculosis classification and detection. The obtained classification accuracy curve, for the detection of TB, was 0.9966 and 0.9978 for both the training and validation sets. The performance of the proposed method can be a very useful and fast diagnostic tool for computer-aided diagnosis using radio-graphs. In the future, the focus will be on reinforcement learning algorithms to classify images that are difficult to classify in a consistent manner.

Conclusion

The Internet of Things (IoT) has drastically transformed the modern digital landscape by connecting an ever-growing number of devices, ranging from household appliances to industrial machines. This interconnectivity has led to an exponential increase in the amount of data generated, processed, and transmitted across networks. While IoT has provided numerous benefits, such as enhanced automation, real-time monitoring, and improved operational efficiency, it has also introduced significant challenges, particularly in the realm of data security and privacy. With the proliferation of IoT devices, ensuring that sensitive information remains secure while minimizing the overhead required to manage such vast data streams has become a critical issue.

At the heart of the IoT security dilemma is the need for lightweight, efficient, and scalable security mechanisms. The inherent resource constraints of IoT devices, such as limited processing power, memory, and energy resources, make traditional security techniques unsuitable for direct application in IoT ecosystems. These limitations necessitate the development of novel approaches that balance the stringent security requirements with the operational limitations of IoT devices. As IoT continues to expand into critical domains like healthcare, industry, and smart cities, ensuring robust security while maintaining system performance is paramount.

This thesis focused on adapting data and communication security techniques specifically tailored to the unique constraints and demands of IoT environments. In traditional computing and communication contexts, security protocols often rely on complex and resource-intensive methods, such as public-key encryption and multi-layered authentication systems. While effective in more conventional networks, these methods can overwhelm IoT devices, which operate with limited computational capacity. Therefore, there is a growing need for lightweight encryption and data integrity solutions that provide robust protection without imposing excessive computational burdens.

To address these challenges, this work contributed to the field of IoT security in two major areas: encryption efficiency and the application of deep learning models to IoT related health data. These two aspects represent critical challenges in IoT, where both data protection and intelligent data analysis are of increasing importance.

The first major contribution of this thesis lies in the development of a novel encryption method that significantly improves data storage and transmission efficiency while maintaining strong security guarantees. Specifically, we introduced the Multi-Key Embedded Encryption (MKEE) technique, which is designed to optimize the ciphertext size, a key challenge in resource-constrained IoT environments. MKEE addresses the problem of data storage by allowing multiple plaintext values to be consolidated into a single ciphertext, thus reducing the overhead associated with securing large datasets.

When combined with the Modular Multiplicative Inverse (MMI) method, this approach achieved remarkable results in terms of data compression and security optimization. The MMI method, which is rooted in number theory, plays a crucial role in further reducing the size of the ciphertext without compromising the integrity or confidentiality of the original data. Our experiments, conducted on a diverse set of IoT-relevant datasets, demonstrated that the MKEE method could achieve an impressive 88% reduction in data storage requirements by consolidating ten plaintext values into a single ciphertext. This is particularly significant in environments where storage capacity is limited and data generation is constant, as is typical in IoT ecosystems.

Furthermore, by enhancing MKEE with the MMI method, we were able to achieve a 94% reduction in storage size when applied to a dataset containing diabetes-related data. This represents a significant advancement in the field of IoT security, as reducing the storage size of encrypted data directly correlates with reduced transmission times, lower energy consumption, and decreased costs associated with data storage. Such improvements are crucial for enabling scalable IoT deployments in healthcare and other industries where data security and efficiency are critical.

Our findings indicate that MKEE, in conjunction with MMI, presents a viable solution for secure data storage in IoT environments, where devices must operate under stringent resource constraints. Future research could explore further refinements to the MKEE method, perhaps incorporating additional optimization techniques from fields such as quantum computing or homomorphic encryption to enhance both the efficiency and security of IoT data.

In addition to encryption, this thesis also explored the application of advanced deep learning techniques to the analysis of IoT-generated health data. The intersection of IoT and healthcare represents one of the most promising areas for improving patient care, diagnostic accuracy, and early detection of diseases. The massive volume of data generated by IoT-enabled medical devices, such as wearable sensors, imaging systems, and remote monitoring devices, offers a wealth of opportunities for machine learning algorithms to uncover insights that might otherwise go unnoticed by traditional diagnostic methods.

This work specifically focused on the development of a deep learning model aimed at improving the early detection of Tuberculosis (TB). TB remains a major global health issue, particularly in low-resource regions where access to trained medical professionals and diagnostic equipment is limited. By leveraging the power of Convolutional Neural Networks (CNNs), we sought to create a system capable of analyzing high-resolution chest X-ray images with a high degree of accuracy.

Our model, which utilized pre-trained CNN architectures such as VGG16 and VGG19, was further enhanced by the incorporation of an attention module. Attention mechanisms have proven to be highly effective in improving the performance

of deep learning models, particularly in tasks that require fine-grained analysis of large datasets. In this context, the attention module allowed the model to focus on the most relevant parts of the chest X-ray image, thereby improving its ability to accurately detect signs of TB.

The model was trained on four large datasets, encompassing a diverse range of TB cases, and achieved near-perfect accuracy, with a training accuracy of 0.9966 and a validation accuracy of 0.9978. These results demonstrate the potential of AI-driven diagnostic systems to augment traditional diagnostic methods, enabling faster and more accurate detection of diseases like TB. In resource-constrained environments, such a system could drastically improve early detection rates and reduce the burden on healthcare systems.

In conclusion, this thesis has made significant contributions to the fields of IoT security and machine learning, with a particular focus on applications in healthcare. The development of the MKEE and MMI methods provides a promising avenue for addressing the storage and transmission challenges inherent in IoT environments. At the same time, the application of deep learning models to healthcare data offers a glimpse into the future of AI-powered diagnostics, where machines assist healthcare professionals in making more accurate and timely decisions.

Future work could extend the methods introduced in this thesis to other IoT applications, particularly in areas such as smart cities, industrial automation, and environmental monitoring. Additionally, further research into the optimization of encryption techniques and the broadening of AI applications across other diseases and datasets could help unlock the full potential of IoT in delivering secure, scalable, and efficient solutions for a wide range of societal challenges.

By continuing to refine these techniques, the Internet of Things can fulfill its promise of enhancing connectivity, security, and societal benefit in a rapidly evolving digital world.

Bibliography

Bibliography

[1] Andrew S. Tanenbaum and Maarten Van Steen. *Distributed Systems: Principles and Paradigms*. Prentice Hall PTR, USA, 1st edition, 2001.

[2] Dieter Gollmann. *Computer Security*. Wiley, 3rd edition, 2011.

[3] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of things: A survey on enabling technologies, protocols and applications. *IEEE Communications Surveys amp Tutorials*, 17:Fourthquarter 2015, 11 2015.

[4] Yan Zhang, Rong Yu, Maziar Nekovee, Yi Liu, Shengli Xie, and Stein Gjessing. Cognitive machine-to-machine communications: visions and potentials for the smart grid. *IEEE Network*, 26(3):6–13, 2012.

[5] Hany F. Atlam, Robert J. Walters, and Gary B. Wills. Fog computing and the internet of things: A review. *Big Data and Cognitive Computing*, 2(2), 2018.

[6] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013. Including Special sections: Cyber-enabled Distributed Computing for Ubiquitous Cloud and Network Services Cloud Computing and Scientific Applications — Big Data, Scalable Analytics, and Beyond.

[7] Igor Gvero. Computers as components, 3rd edition: principles of embedded computing system design by marilyn wolf. *SIGSOFT Softw. Eng. Notes*, 38(5):67–68, aug 2013.

[8] John A. Stankovic. Research directions for the internet of things. *IEEE Internet of Things Journal*, 1(1):3–9, 2014.

[9] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. Internet of things for smart cities. *IEEE Internet of Things Journal*, 1(1):22–32, 2014.

[10] S. M. 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.

[11] Rolf Weber and Evelyne Studer. Cybersecurity in the internet of things: Legal aspects. *Computer Law Security Review*, 32, 08 2016.

[12] Min Chen, Shiwen Mao, and Yunhao Liu. Big data: A survey. *Mobile Networks and Applications*, 19(2):171–209, 2014.

[13] Ibrahim Hashem, Ibrar Yaqoob, Nor Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Khan. The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, 47:98–115, 07 2014.

[14] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597, 2016.

[15] Alaba Fadele, Mazliza Othmana, Ibrahim Hashem, and Faiz Alotaibi. Internet of things security: A survey. *Journal of Network and Computer Applications*, 88, 04 2017.

[16] 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.

[17] Rodrigo Roman, Javier Lopez, and Masahiro Mambo. Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems*, 78:680–698, 2018.

[18] Xing Hu, Ge Li, Xin Xia, David Lo, Shuai Lu, and Zhi Jin. Summarizing source code with transferred api knowledge. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 2269–2275. AAAI Press, 2018.

[19] Romaissa Kebache, Abdelkader Laouid, Ahcene Bounceur, Mostefa Kara, Konstantinos Karampidis, Giorgos Papadourakis, and Mohammad Hammoudeh. Reducing the encrypted data size: Healthcare with iot-cloud computing applications. *Computer Systems Science & Engineering*, 48(4), 2024.

[20] Mostefa Kara, Abdelkader Laouid, Ahcene Bounceur, Mohammad Hammoudeh, Muath Alshaikh, and Romaissa Kebache. Semi-decentralized model for drone collaboration on secure measurement of positions. In *Proceedings of the 5th International Conference on Future Networks and Distributed Systems*, pages 64–69, 2021.

[21] Ravi Singh, Mohd Javaid, Abid Haleem, and Rajiv Suman. Internet of things (iot) applications to fight against covid-19 pandemic. *Diabetes Metabolic Syndrome: Clinical Research Reviews*, 14, 05 2020.

[22] Mark Hung. Leading the iot, gartner insights on how to lead in a connected world. *Gartner Research*, pages 1–29, 2017.

[23] Horst Stipp. Statista Research Departement internet of things- number of connected devices worldwide 2015 – 2025. `https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/`, 2016. Accessed: 2020-07-26.

[24] Knud Lasse Lueth. State of the IoT 2018 number of iot devices now at 7b – market accelerating. `https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/`, 2018. Accessed: 2020-07-26.

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

[26] Sufian Hameed, Faraz Idris Khan, and Bilal Hameed. Understanding security requirements and challenges in internet of things (iot): A review. *Journal of Computer Networks and Communications*, 2019, 2019.

[27] Alekha Parimal Bhatt and Anand Sharma. Quantum cryptography for internet of things security. *Journal of Electronic Science and Technology*, 17(3):213–220, 2019.

[28] Jyoti Deogirikar and Amarsinh Vidhate. Security attacks in iot: A survey. In *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pages 32–37. IEEE, 2017.

[29] Xu Wang, Xuan Zha, Wei Ni, Ren Liu, Y. Guo, Xinxin Niu, and Kangfeng Zheng. Survey on blockchain for internet of things. *Computer Communications*, 136, 01 2019.

[30] Hezam Akram Abdul-Ghani, Dimitri Konstantas, and Mohammed Mahyoub. A comprehensive iot attacks survey based on a building-blocked reference model. *IJACSA) International Journal of Advanced Computer Science and Applications*, 9(3):355–373, 2018.

[31] B. Karg and S. Lucia. Towards low-energy, low-cost and high-performance iot-based operation of interconnected systems. In *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pages 706–711, 2018.

[32] Pallavi Sethi and Smruti R Sarangi. Internet of things: architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, 2017, 2017.

[33] B. Cao, Y. Li, L. Zhang, L. Zhang, S. Mumtaz, Z. Zhou, and M. Peng. When internet of things meets blockchain: Challenges in distributed consensus. *IEEE Network*, 33(6):133–139, 2019.

[34] Pierre-Alain Fouque. *Le partage de clés cryptographiques: Théorie et Pratique*. PhD thesis, Paris 7, 2001.

[35] Yogesh Kumar, Rajiv Munjal, and Harsh Sharma. Comparison of symmetric and asymmetric cryptography with existing vulnerabilities and countermeasures. *International Journal of Computer Science and Management Studies*, 11, 10 2011.

[36] Monica Pawlan. Cryptography: The ancient art of secret messages. `http://www.pawlan.com/monica/articles/crypto/`, February 1998. Accessed: 2009-05-04.

[37] Michael E. Whitman and Herbert J. Mattord. *Principles of Information Security*. Thomson Learning, Inc., Canada, 2005. [University of Phoenix Custom Edition e-text].

[38] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[39] Elisabeth Rakus-Andersson. *The Polish Brains Behind the Breaking of the Enigma Code Before and During the Second World War*, pages 419–439. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[40] Suhri Kim and Seokhie Hong. Design and security analysis of cryptosystems. *Applied Sciences*, 13(3):1427, 2023.

[41] Zenith Dewamuni, Bharanidharan Shanmugam, Sami Azam, and Suresh Thennadil. Bibliometric analysis of iot lightweight cryptography. *Information*, 14(12), 2023.

[42] William J Buchanan and Leandros Maglaras. Review of the nist light-weight cryptography finalists, 2023.

[43] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2: Lightweight authenticated encryption and hashing. *Journal of Cryptology*, 34:33, 2021.

[44] Subhadeep Banik, Atul Luykx, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. Gift-cofb. `https://eprint.iacr.org/2020/738`, 2020. Cryptology ePrint Archive, Paper 2020/738.

[45] Martin Hell, Thomas Johansson, Alexander Maximov, and Willi Meier. Grain-128aead v1.0. `https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/Grain_128AEAD-spec.pdf`, 2019. NIST Lightweight Cryptography Project.

[46] Zhenzhen Bao, Abhijit Chakraborti, and Yu Sasaki. Photon-beetle v1.0. `https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/PHOTON-Beetle-spec.pdf`, 2019. NIST Lightweight Cryptography Project.

[47] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Isap v2.0. `https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/isap-spec-final.pdf`, 2019. NIST Lightweight Cryptography Project.

[48] Joseph Jaeger and Akshaya Kumar. Memory-tight multi-challenge security of public-key encryption. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology – ASIACRYPT 2022*, pages 454–484, Cham, 2022. Springer Nature Switzerland.

[49] Yi-Kai Liu and Dustin Moody. Post-quantum cryptography and the quantum future of cybersecurity. *Phys. Rev. Appl.*, 21:040501, Apr 2024.

[50] Ranjan Kumar Thakur, Nawin Kumar Agrawal, and P. Kumar. A symmetric key cryptosystem based on singular linear transformation. *AIP Conference Proceedings*, 2852(1):060001, 08 2023.

[51] T. Jamil. The rijndael algorithm. *IEEE Potentials*, 23(2):36–38, 2004.

[52] National Bureau of Standards. Data Encryption Standard (DES). Technical Report FIPS PUB 46, National Bureau of Standards, 1977. `https://csrc.nist.gov/pubs/fips/46/final`.

[53] Xuejia Lai and James L. Massey. A proposal for a new block encryption standard. In *Advances in Cryptology — EUROCRYPT '90*, volume 473 of *Lecture Notes in Computer Science*, pages 389–404. Springer, 1991.

[54] Bruce Schneier. Description of a new variable-length key, 64-bit block cipher (blowfish). In *Fast Software Encryption*, volume 809 of *Lecture Notes in Computer Science*, pages 191–204. Springer, 1993.

[55] Saif Al-Kuwari, James H Davenport, and Russell J Bradford. Cryptographic hash functions: Recent design trends and security notions. *Cryptology ePrint Archive*, 2011.

[56] Takanori Isobe, Ryoma Ito, Fukang Liu, Kazuhiko Minematsu, Motoki Nakahashi, Kosei Sakamoto, and Rentaro Shiba. Areion: highly-efficient permutations and its applications to hash functions for short input. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 115–154, 2023.

[57] 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.

[58] Sabina Szymoniak and Shalini Kesar. Key agreement and authentication protocols in the internet of things: A survey. *Applied Sciences*, 13(1), 2023.

[59] Ammar Mohammad, Hasan Al-Refai, and Ali Ahmad Alawneh. User authentication and authorization framework in iot protocols. *Computers*, 11(10), 2022.

[60] Mohammad Reza Servati, Masoumeh Safkhani, Saqib Ali, Mazhar Hussain Malik, Omed Hassan Ahmed, Mehdi Hosseinzadeh, and Amir H. Mosavi. Cryptanalysis of two recent ultra-lightweight authentication protocols. *Mathematics*, 10(23), 2022.

[61] Stanislaw Jarecki, Hugo Krawczyk, and Jason Resch. Updatable oblivious key management for storage systems. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 379–393, 2019.

[62] Duc-Thuan Dam, Thai-Ha Tran, Van-Phuc Hoang, Cong-Kha Pham, and Trong-Thuc Hoang. A survey of post-quantum cryptography: Start of a new race. *Cryptography*, 7(3), 2023.

[63] Ritik Bavdekar, Eashan Jayant Chopde, Ashutosh Bhatia, Kamlesh Tiwari, Sandeep Joshua Daniel, and Atul. Post quantum cryptography: Techniques, challenges, standardization, and directions for future research, 2022.

[64] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

[65] Jihoon Cho, Jincheol Ha, Seongkwang Kim, Byeonghak Lee, Joohee Lee, Jooyoung Lee, Dukjae Moon, and Hyojin Yoon. Transciphering framework for approximate homomorphic encryption (full version). *Cryptology ePrint Archive*, 2020.

[66] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. *SIAM Journal on Computing*, 43(2):831–871, 2014.

[67] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Comput. Surv.*, 51(4), jul 2018.

[68] Michele Minelli. *Fully homomorphic encryption for machine learning*. Theses, Université Paris sciences et lettres, October 2018.

[69] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, page 169–178, New York, NY, USA, 2009. Association for Computing Machinery.

[70] Shai Halevi and Victor Shoup. Algorithms in helib. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014*, pages 554–571, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[71] Nigel P Smart and Frederik Vercauteren. Fully homomorphic simd operations. *Designs, codes and cryptography*, 71:57–81, 2014.

[72] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*, CCSW '11, page 113–124, New York, NY, USA, 2011. Association for Computing Machinery.

[73] Kim Laine. Simple encrypted arithmetic library 2.3. 1. *Microsoft Research https://www. microsoft. com/en-us/research/uploads/prod/2017/11/sealmanual-2-3-1. pdf*, 2017.

[74] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, pages 409–437, Cham, 2017. Springer International Publishing.

[75] Wenyi Tang1, Bo Qin, Yanan Li, and Qianhong Wu and. Functional privacy-preserving outsourcing scheme with computation verifiability in fog computing. *KSII Transactions on Internet and Information Systems*, 14(1):281–298, January 2020.

[76] Yanwei Gong, Xiaolin Chang, Jelena Mišić, Vojislav B Mišić, Jianhua Wang, and Haoran Zhu. Practical solutions in fully homomorphic encryption: a survey analyzing existing acceleration methods. *Cybersecurity*, 7(1):5, 2024.

[77] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019.

[78] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference*

on Computer Vision and Pattern Recognition (CVPR), pages 4396–4405, 2019.

[79] Ziadoon K Maseer, Robiah Yusof, Salama A Mostafa, Nazrulazhar Bahaman, Omar Musa, and B Ali Saleh Al-rimy. Deepiot. ids: Hybrid deep learning for enhancing iot network intrusion detection. *Comput. Mater. Contin*, 69(3):3945–3966, 2021.

[80] Fatima Hussain, Rasheed Hussain, Syed Ali Hassan, and Ekram Hossain. Machine learning in iot security: Current solutions and future challenges. *IEEE Communications Surveys Tutorials*, 22(3):1686–1721, 2020.

[81] Cristiano Antonio de Souza, Carlos Becker Westphall, Renato Bobsin Machado, João Bosco Mangueira Sobral, and Gustavo dos Santos Vieira. Hybrid approach to intrusion detection in fog-based iot environments. *Computer Networks*, 180:107417, 2020.

[82] Abhijit Bhattacharyya, Divyanshu Bhaik, Sunil Kumar, Prayas Thakur, Rahul Sharma, and Ram Bilas Pachori. A deep learning based approach for automatic detection of covid-19 cases using chest x-ray images. *Biomedical Signal Processing and Control*, 71:103182, 2022.

[83] Ziad E. Dawahdeh, Shahrul N. Yaakob, and Rozmie Razif bin Othman. A new image encryption technique combining elliptic curve cryptosystem with hill cipher. *Journal of King Saud University - Computer and Information Sciences*, 30(3):349–355, 2018.

[84] Falguni Patel and Mohammed Farik. A new substitution cipher -random-x. *International Journal of Scientific Technology Research*, 5:125–128, 11 2016.

[85] Porter E. Coggins III and Tim Glatzer. An algorithm for a matrix-based enigma encoder from a variation of the hill cipher as an application of 2 $\times$ 2 matrices. *PRIMUS*, 30(1):1–18, 2020.

[86] D.R. Lide, D.R. Lide, National Institute of Standards, and Technology (U.S.). *A Century of Excellence in Measurements, Standards, and Technology 1901-2000*. NIST special publication. U.S. Department of Commerce, National Institute of Standards and Technology, 2001.

[87] Nan Li. Research on diffie-hellman key exchange protocol. In *2010 2nd International Conference on Computer Engineering and Technology*, volume 4, pages V4–634–V4–637, 2010.

[88] Ronald L. Rivest and Michael L. Dertouzos. On data banks and privacy homomorphisms. 1978.

[89] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Comput. Surv.*, 51(4), jul 2018.

[90] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, page 169–178, New York, NY, USA, 2009. Association for Computing Machinery.

[91] 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.

[92] James Dyer, Martin Dyer, and Jie Xu. Practical homomorphic encryption over the integers for secure computation in the cloud. *International Journal of Information Security*, 18, 10 2019.

[93] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[94] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In Joe Kilian, editor, *Theory of Cryptography*, pages 325–341, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[95] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, pages 24–43, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[96] Smaranika Dasgupta and S.K. Pal. Design of a polynomial ring based symmetric homomorphic encryption scheme. *Perspectives in Science*, 8:692–695, 2016. Recent Trends in Engineering and Material Sciences.

[97] Yarkın Doröz, Aria Shahverdi, Thomas Eisenbarth, and Berk Sunar. Toward practical homomorphic evaluation of block ciphers using prince. In Rainer Böhme, Michael Brenner, Tyler Moore, and Matthew Smith, editors, *Financial Cryptography and Data Security*, pages 208–220, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[98] Yarkın Doröz, Aria Shahverdi, Thomas Eisenbarth, and Berk Sunar. Toward practical homomorphic evaluation of block ciphers using prince. In Rainer Böhme, Michael Brenner, Tyler Moore, and Matthew Smith, editors, *Financial Cryptography and Data Security*, pages 208–220, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[99] Mostefa Kara, Abdelkader Laouid, Ahcene Bounceur, Farid Lalem, Muath AlShaikh, Romaissa Kebache, and Zaoui Sayah. A novel delegated proof of work consensus protocol. In *2021 International Conference on Artificial Intelligence for Cyber Security Systems and Privacy (AI-CSP)*, pages 1–7. IEEE, 2021.

[100] Aissaoua Habib, Abdelkader Laouid, and Mostefa Kara. Secure consensus clock synchronization in wireless sensor networks. In *2021 International Conference on Artificial Intelligence for Cyber Security Systems and Privacy (AI-CSP)*, pages 1–6. IEEE, 2021.

[101] Ning Hu, Zhihong Tian, Xiaojiang Du, Nadra Guizani, and Zhihan Zhu. Deep-green: A dispersed energy-efficiency computing paradigm for green industrial iot. *IEEE Transactions on Green Communications and Networking*, 5(2):750–764, 2021.

[102] Chang Liu, Gang Xiong, Gaopeng Gou, Siu-Ming Yiu, Zhen Li, and Zhihong Tian. Classifying encrypted traffic using adaptive fingerprints with multi-level attributes. *World Wide Web*, 24, 11 2021.

[103] S. P. Chokkalingam E. Srimathi. Improved cloud storage encryption using block cipher-based dna anti-codify model. *Computer Systems Science and Engineering*, 47(1):903–918, 2023.

[104] Naglaa F. Soliman Abeer D. Algarni Fathi E. Abd El-Samie Walid El-Shafai 4 Musheer Ahmad, Reem Ibrahim Alkanhel. Securing healthcare data in iomt network using enhanced chaos based substitution and diffusion. *Computer Systems Science and Engineering*, 47(2):2361–2380, 2023.

[105] Liang Wei, Xu Jianbo, Tang Mingdong, and Huang Li. A new embedded encryption algorithm for wireless sensor networks. In *2009 International Forum on Information Technology and Applications*, volume 1, pages 119–122, 2009.

[106] Mohammadi R., Salehi M, Ghaffari H., Rohani A. A, and Reiazi R. Transfer learning-based automatic detection of coronavirus disease 2019 (covid-19) from chest x-ray images. *Journal of Biomedical Physics & Engineering*, 10:559 – 568, 2020.

[107] Yibo Feng, Xu Yang, Dawei Qiu, Huan Zhang, Dejian Wei, and Jing Liu. Pcxrnet: Pneumonia diagnosis from chest x-ray images using condense attention block and multiconvolution attention block. *IEEE Journal of Biomedical and Health Informatics*, 26(4):1484–1495, 2022.

[108] Tawsifur Rahman, Amith Khandakar, Muhammad Abdul Kadir, Khandaker Rejaul Islam, Khandakar F. Islam, Rashid Mazhar, Tahir Hamid, Mohammad Tariqul Islam, Saad Kashem, Zaid Bin Mahbub, Mohamed Arselene

Ayari, and Muhammad E.H. Chowdhury. Reliable tuberculosis detection using chest x-ray with deep learning, segmentation and visualization. *IEEE Access*, 8:191586–191601, 2020.

[109] Sonaal Kant and Muktabh Mayank Srivastava. Towards automated tuberculosis detection using deep learning. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1250–1253, 2018.

[110] Rahul Hooda, Sanjeev Sofat, Simranpreet Kaur, Ajay Mittal, and Fabrice Meriaudeau. Deep-learning: A potential method for tuberculosis detection using chest radiography. In *2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 497–502, 2017.

[111] R. Dinesh, Jackson Samuel, • B. Rajesh Kanna, and B. Rajesh Kanna. Tuberculosis (tb) detection system using deep neural networks. *Neural Computing and Applications*, 31:1533 – 1545, 2018.

[112] Stefanus Hwa, Abdullah Bade, Mohd Hijazi, and Mohammad Jeffree. Tuberculosis detection using deep learning and contrastenhanced canny edge detected x-ray images. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 9:713, 12 2020.

[113] Ruihua Guo, Kalpdrum Passi, and Chakresh Kumar Jain. Tuberculosis diagnostics and localization in chest x-rays via deep learning models. *Frontiers in Artificial Intelligence*, 3:583427, 2020.

[114] Mustapha Oloko-Oba and Serestina Viriri. A systematic review of deep learning techniques for tuberculosis detection from chest radiograph. *Frontiers in Medicine*, 9, 2022.

[115] Mahantesh N. Birje, Praveen S. Challagidad, R.H. Goudar, and Manisha T. Tapale. Cloud computing review: concepts, technology, challenges and security. *International Journal of Cloud Computing*, 6(1):32–57, 2017.

[116] Robert C. Blattberg, Byung-Do Kim, and Scott A. Neslin. *Market Basket Analysis*, pages 339–351. Springer New York, New York, NY, 2008.

[117] Olcay Genc, Atıl Kurt, Devrim Murat Yazan, and Ercan Erdis. Circular eco-industrial park design inspired by nature: An integrated non-linear optimization, location, and food web analysis. *Journal of Environmental Management*, 270:110866, 2020.

[118] Chonho Lee, Zhaojing Luo, Kee Yuan Ngiam, Meihui Zhang, Kaiping Zheng, Gang Chen, Beng Chin Ooi, and Wei Luen James Yip. *Big Healthcare Data Analytics: Challenges and Applications*, pages 11–41. Springer International Publishing, Cham, 2017.

[119] Kalyan Nagaraj, G.S. Sharvani, and Amulyashree Sridhar. Emerging trend of big data analytics in bioinformatics: a literature review. *International Journal of Bioinformatics Research and Applications*, 14(1-2):144–205, 2018.

[120] Luca Oneto, Emanuele Fumeo, Giorgio Clerico, Renzo Canepa, Federico Papa, Carlo Dambra, Nadia Mazzino, and Davide Anguita. Train delay prediction systems: A big data analytics perspective. *Big Data Research*, 11:54–64, 2018. Selected papers from the 2nd INNS Conference on Big Data: Big Data Neural Networks.

[121] Mehdi Mohammadpoor and Farshid Torabi. Big data analytics in oil and gas industry: An emerging trend. *Petroleum*, 6(4):321–328, 2020. SI: Artificial Intelligence (AI), Knowledge-based Systems (KBS), and Machine Learning (ML).

[122] N. MuthuLakshmi and K. Rani. Privacy preserving association rule mining in vertically partitioned databases. *International Journal of Computer Applications*, 39:29–35, 02 2012.

[123] Khaled Chait, Mostefa Kara, Abdelkader Laouid, Mohammad Hammoudeh, and Ahcène Bounceur. One digit checksum for data integrity verification of cloud-executed homomorphic encryption operations. In *Proceedings of the 7th International Conference on Future Networks and Distributed Systems*, pages 71–75, 2023.

[124] 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), 2023.

[125] M. Iqbal Bhat and Kaiser J. Giri. *Impact of Computational Power on Cryptography*, pages 45–88. Springer Singapore, Singapore, 2021.

[126] 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), 2023.

[127] Mostefa Kara, Konstantinos Karampidis, Giorgos Papadourakis, Abdelkader Laouid, and Muath AlShaikh. A probabilistic public-key encryption with ensuring data integrity in cloud computing. In *2023 International Conference on Control, Artificial Intelligence, Robotics Optimization (ICCAIRO)*, pages 59–66, 2023.

[128] Mohammad Hammoudeh, Gregory Epiphaniou, Sana Belguith, Devrim Unal, Bamidele Adebisi, Thar Baker, A. S. M. Kayes, and Paul Watters.

A service-oriented approach for sensing in the internet of things: Intelligent transportation systems and privacy use cases. *IEEE Sensors Journal*, 21(14):15753–15761, 2021.

[129] Mohiuddin Ahmed and Abu S. S. M. Barkat Ullah. False data injection attacks in healthcare. In Yee Ling Boo, David Stirling, Lianhua Chi, Lin Liu, Kok-Leong Ong, and Graham Williams, editors, *Data Mining*, pages 192–202, Singapore, 2018. Springer Singapore.

[130] Adil Hussain Seh, Mohammad Zarour, Mamdouh Alenezi, Amal Krishna Sarkar, Alka Agrawal, Rajeev Kumar, and Raees Ahmad Khan. Healthcare data breaches: Insights and implications. *Healthcare*, 8(2), 2020.

[131] Ramzi Haraty, Mirna Zbib, and Mehedi Masud. Data damage assessment and recovery algorithm from malicious attacks in healthcare data sharing systems. *Peer-to-Peer Networking and Applications*, 9, 05 2015.

[132] Nuttapong Attrapadung, Javier Herranz, Fabien Laguillaumie, Benoît Libert, Elie de Panafieu, and Carla Ràfols. Attribute-based encryption schemes with constant-size ciphertexts. *Theoretical Computer Science*, 422:15–38, 2012.

[133] Muhammad Usman and Shujaat Khan. Sit: A lightweight encryption algorithm for secure internet of things. *International Journal of Advanced Computer Science and Applications*, 8, 02 2017.

[134] Yaser Mansouri, Adel Toosi, and Rajkumar Buyya. Brokering algorithms for optimizing the availability and cost of cloud storage services. volume 1, pages 581–589, 12 2013.

[135] Robbi Rahim and Ali Ikhwan. Cryptography technique with modular multiplication block cipher and playfair cipher. *International Journal of Scientific Research in Science and Technology*, 2:71–78, 11 2016.

[136] Ndibanje Bruce, Won Tae Jang, and Hoon Jae Lee. An embedded encryption protocol for healthcare networks security. *International Journal of Security and Its Applications*, 8(2):139–144, 2014.

[137] Yunusa Simpa Abdulsalam and Mustapha Hedabou. Security and privacy in cloud computing: Technical review. *Future Internet*, 14(1), 2022.

[138] John Dawson, Frimpong Twum, James Acquah, and Yaw Missah. Ensuring confidentiality and privacy of cloud data using a non-deterministic cryptographic scheme. *PloS one*, 18:e0274628, 02 2023.

[139] Kristian F Andvord. What can we learn by following the development of tuberculosis from one generation to another?[founders of our knowledge]. *The International Journal of Tuberculosis and Lung Disease*, 6(7):562–568, 2002.

[140] Kara Mostefa, Konstantinos Karampidis, Zaoui Sayah, Abdelkader Laouid, Giorgos Papadourakis, and Mohamed Nadhir Abid. *A Password-Based Mutual Authentication Protocol via Zero-Knowledge Proof Solution*, pages 31–40. 09 2023.

[141] Khaled Chait, Abdelkader Laouid, Mostefa Kara, Mohammad Hammoudeh, Omar Aldabbas, and Abdullah T Al-Essa. An enhanced rsa-based aggregate signature scheme to reduce blockchain size. *IEEE Access*, 2023.

[142] Kara Mostefa, Konstantinos Karampidis, Giorgos Papadourakis, Abdelkader Laouid, and Muath Alshaikh. A probabilistic public-key encryption with ensuring data integrity in cloud computing. pages 59–66, 04 2023.

[143] 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.

[144] Chiranjibi Sitaula and Mohammad Hossain. Attention-based vgg-16 model for covid-19 chest x-ray image classification. *Applied Intelligence*, 51:1–14, 05 2021.

[145] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[146] Srikanth Tammina. Transfer learning using vgg-16 with deep convolutional neural network for classifying images. volume 9, page p9420, 10 2019.

[147] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.