UNIVERSITY MOHAMED KHIDER BISKRA
Faculty of Exact Sciences and Natural and Life Sciences
DEPARTMENT OF COMPUTER SCIENCE

DOCTORAT THESIS

In order to obtain the degree of **LMD Doctorate** in computer science

Option : Networking and Distributed Systems

Title

Towards Green Security for E-health Applications in The Internet of Things

Presented by : Meriem GASMI

Defended on: **26/06/2025**, in front of the jury:

Mr. CHERIF Foudil	Professeur	University of Biskra	Chairperson
Mr. Abdelmalik BACHIR	Professor	ENSIA	Supervisor
Mr. CHERIET Abdelhakim	MCA	ENSIA	Examiner
Mr. ZERARKA Mohamed Faouzi	Professeur	University of Biskra	Examiner

Dedication

I would like to dedicate this thesis to my mother,
My sisters and brothers,
My husband and daughter,
For their unwavering support, patience, and encouragement.
And in loving memory of my dear father, who will forever hold a special place in my heart.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to Allah (God), the Most Gracious and the Most Merciful, for granting me the strength, patience, and perseverance to embark on and complete this PhD journey.

I would like to express my deep gratitude to my supervisor, Professor Abdelmalik Bachir, for his support and encouragement throughout my PhD journey. His expertise and dedication were pivotal in the completion of this research.

A special thanks to my co-supervisor, Doctor Kerdoudi Mohamed Lamine, for his invaluable insights and thoughtful guidance, which have been indispensable throughout this process. His constructive feedback, high attention to detail, and willingness to share his deep knowledge have significantly enriched my research and helped me overcome various challenges.

I am also profoundly thankful to the members of the LESIA laboratory, especially Mrs. A. Rahim and Professor Foudil Cherif, for their encouragement.

I would like to sincerely thank the members of the jury for accepting the invitation to be part of this significant moment. Your time, effort, and valuable insights are greatly appreciated, and I am honored by your participation in the evaluation of my work.

On a personal note, to my mother, thank you for your unwavering belief in me, your never-ending encouragement, and for being my constant pillar of strength. Your unconditional love, immense patience, and selfless support have been a source of comfort and motivation throughout this journey. I am deeply grateful for the sacrifices you have made which have fueled my determination to succeed. To my brothers and sisters, thank you for being my source of resilience and for always cheering me on from afar. I thank my siblings for their encouragement, my husband, Abou Bakr Seddik, for his steadfast support and patience, and my daughter, Israa, for being a constant source of joy and inspiration.

I extend my heartfelt appreciation to all my friends, both near and far, for their support and motivation, which have helped me stay strong throughout this journey.

Thank you all for your invaluable contributions to this achievement.

Abstract

The Internet of Things (IoTs) is an emerging technology that connects various devices and systems to the Internet, allowing them to communicate and share data. This interconnected network of devices has the potential to revolutionize industries, improve efficiency, and enhance our daily lives. The IoT relies on physical sensors that gather data and transmit it to remote cloud computing platforms for analysis and storage. One of the most promising fields for IoT applications is healthcare, where significant advancements are anticipated. For instance, devices can be implanted in patients' bodies to monitor vital signs, transmitting the data to the cloud for storage, processing, and informed decision-making. The data should then be accessed by healthcare professionals in real-time, allowing for immediate intervention when necessary. With the potential to revolutionize the healthcare industry, the integration of IoT technology in healthcare applications is becoming increasingly prevalent and essential. This research focuses on security aspects, specifically addressing the confidentiality and access control of data (e.g., EMR) during transmission and storage in the cloud. In addition, we examine the energy consumption of IoT devices in healthcare applications, ensuring their optimization for better efficiency. To ensure the confidentiality and security of these sensitive data, encryption techniques and access control protocols must be carefully implemented. Energy efficiency is another crucial aspect to consider, as IoT devices in healthcare applications must be able to operate continuously and autonomously without draining excessive power. By addressing these challenges, we can fully harness the potential of IoT technology to improve patient outcomes and streamline healthcare delivery. Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is widely regarded as an ideal method for implementing fine-grained access control. However, existing CP-ABE solutions are not efficient or well-suited for the IoT environment, as data producers are typically highly resource-constrained and unable to perform the public-key cryptographic functions required by CP-ABE. In this thesis, we suggest enhancing the standard CP-ABE to make it more suitable for resource-constrained sensors by offloading the intensive encryption tasks to multiple cooperative nodes. This approach ensures a balanced workload distribution between the sensors and the assistant nodes, taking into account the capacity of the assistant nodes for optimal distribution. We evaluated our proposal through a series of experiments, and the results demonstrate that, compared to existing outsourcing ABE solutions, our approach significantly improves

both computation time and energy efficiency, guarantees data confidentiality, prolongs the sensor's battery life, and ensures fine-grained access control.

Keywords: IoT, Security, Fine access control, CP-ABE, Outsourced ABE, Load-balancing, Energy consumption.

Résumé

L'Internet des objets (IoT) est une technologie émergente qui permet de connecter différents appareils et systèmes à Internet, leur permettant de communiquer et de partager des données. Ce réseau interconnecté d'appareils a le potentiel de révolutionner les industries, d'améliorer l'efficacité et d'enrichir notre quotidien. L'IoT est basé sur des capteurs physiques qui recueillent des données et les transfèrent vers des plateformes de cloud à distance pour le traitement et le stockage. Les applications de santé sont l'un des domaines très prometteurs où l'IoT devrait faire des progrès significatifs. En effet, des dispositifs peuvent être installés dans le corps des patients pour surveiller les paramètres vitaux et envoyer les données dans le cloud pour le stockage, le traitement et la prise de décisions. Ces données devraient ensuite être accessibles en temps réel par les experts de la santé, permettant une intervention immédiate si nécessaire. Avec le potentiel de révolutionner l'industrie de la santé, l'intégration de la technologie IoT dans les applications de santé devient de plus en plus répandue et essentielle. Cette recherche se concentre sur les aspects de sécurité, en particulier sur la confidentialité et le contrôle d'accès des données (par exemple, EMRs) lors de la transmission et du stockage dans le cloud. En outre, nous examinons la consommation d'énergie des dispositifs IoT dans les applications de santé, en veillant à leur optimisation pour une meilleure efficacité. Pour assurer la confidentialité et la sécurité de ces données sensibles, les techniques de chiffrement et les protocoles de contrôle d'accès doivent être soigneusement mis en œuvre. L'efficacité énergétique est un autre aspect crucial à prendre en considération, car les dispositifs IoT dans les applications de santé doivent pouvoir fonctionner en continu et de manière autonome sans gaspiller excessivement l'énergie. En abordant ces défis, nous pouvons exploiter pleinement le potentiel de la technologie IoT pour améliorer les résultats des patients et rationaliser la prestation des soins de santé. Le chiffrement basé sur les attributs de la politique de chiffrement (CP-ABE) est considéré comme le plus pratique pour la réalisation du contrôle d'accès fin. En fait, les solutions actuelles de CP-ABE ne sont pas efficaces et ne conviennent pas à l'utilisation dans le contexte de l'Internet des objets, car les producteurs de données sont généralement très restreints et ne peuvent donc pas exécuter les fonctions cryptographiques à clé publique utilisées par CP-ABE. Dans cette thèse, nous proposons d'améliorer la norme CP-ABE pour la rendre plus pratique pour les capteurs restreints en déchargant les calculs de chiffrement lourds sur plusieurs nœuds

coopératifs. Cela permet d'assurer une distribution équitable de la charge de travail entre plusieurs capteurs et plusieurs nœuds coopératifs. Nous nous basons sur la capacité des nœuds assistants pour la meilleure distribution de la charge de travail. Nous avons évalué notre proposition à travers un ensemble d'expériences. Les résultats obtenus montrent que notre approche, par rapport aux solutions ABE outsourcées existantes, améliore le temps de calcul et la consommation d'énergie, assure la confidentialité des données, prolonge l'autonomie de la batterie du capteur et fournit un contrôle d'accès fin.

Mots-clés : IoT, Sécurité, Contrôle d'accès fin, CP-ABE, ABE externalisé, Équilibrage des charges, Consommation d'énergie.

ملخص

إنترنت الأشياء ($IoTs$) هي تقنية ناشئة تربط بين مختلف الأجهزة والأنظمة بالإنترنت، مما يتيح لها التواصل ومشاركة البيانات. هذه الشبكة المترابطة من الأجهزة لديها القدرة على إحداث ثورة في الصناعات، وتحسين الكفاءة، وتعزيز حياتنا اليومية. إنترنت الأشياء يعتمد على أجهزة استشعار مادية تجمع البيانات وتنقلها إلى منصات الحوسبة السحابية البعيدة للمعالجة والتخزين. تُعتبر تطبيقات الرعاية الصحية واحدة من المجالات الواعدة جداً التي يُتوقع أن تحقق فيها إنترنت الأشياء تقدماًا ملحوظاًا. بالفعل، يمكن تركيب أجهزة في أجسام المرضى لمراقبة العلامات الحيوية وإرسال البيانات إلى السحابة للتخزين والمعالجة واتخاذ القرارات. يجب أن يتمكن المتخصصون في الرعاية الصحية من الوصول إلى هذه البيانات في الوقت الفعلي، مما يتيح التدخل الفوري عند الحاجة. بفضل إمكاناتها في إحداث ثورة في قطاع الرعاية الصحية، أصبح دمج تقنية إنترنت الأشياء في التطبيقات الصحية أكثر انتشاراً وأهمية.

يتناول هذا البحث جوانب الأمان، مع التركيز بشكل خاص على سرية البيانات والتحكم في الوصول إليها (مثل السجلات الطبية الإلكترونية) أثناء النقل والتخزين في السحابة. بالإضافة إلى ذلك، نقوم بدراسة استهلاك الطاقة لأجهزة إنترنت الأشياء في تطبيقات الرعاية الصحية، مع ضمان تحسينها لتحقيق كفاءة أفضل. لضمان سرية وأمان هذه البيانات الحساسة، يجب تنفيذ تقنيات التشفير وبروتوكولات التحكم في الوصول بعناية. كفاءة الطاقة هي جانب آخر حاسم يجب أخذه بعين الاعتبار، حيث يجب أن تكون أجهزة إنترنت الأشياء في تطبيقات الرعاية الصحية قادرة على العمل بشكل مستمر ومستقل دون استهلاك طاقة مفرطة. من خلال معالجة هذه التحديات، يمكننا استغلال إمكانيات تكنولوجيا إنترنت الأشياء بشكل كامل لتحسين نتائج المرضى وتبسيط تقديم الرعاية الصحية. يعتبر تشفير النصوص بناءً على تقنية ($CP-ABE$)الأكثر ملاءمة لتحقيق التحكم الدقيق في الوصول. في الواقع، الحلول الحالية لتشفير الوصول القائم على الخصوصية ($CP-ABE$) ليست فعالة ومناسبة للاستخدام في سياق إنترنت الأشياء، حيث أن منتجي البيانات عادة ما يكونون مقيدين جداً وبالتالي لا يمكنهم تنفيذ وظائف التشفير بالمفتاح العام المستخدمة في $CP-ABE$.

في هذه الأطروحة، نقترح تحسين نظام $CP-ABE$ من خلال تحميل عمليات التشفير الثقيلة إلى عدة عقد تعاونية مع ضمان توزيع متوازن لحمولة العمل بين عدة حساسات وعدة عقد تعاونية.

استندنا إلى قدرة عقد المساعدين لتحقيق أفضل توزيع للأحمال. قمنا بتقييم اقتراحنا من خلال مجموعة من التجارب. تظهر النتائج التي تم الحصول عليها أن نهجنا، مقارنةً بحلول ABE التي تعتمد على اخراج عمليات التشفير الثقيلة الحالية، يحسن من وقت الحساب واستهلاك الطاقة، ويضمن سرية البيانات، ويطيل عمر بطارية المستشعر، ويوفر تحكمًا دقيقًا في الوصول.

الكلمات المفتاحية: إنترنت الأشياء، الأمن، التحكم الدقيق في الوصول، $CP-ABE$، ABE المُعتمد على الخدمات الخارجية، توزيع الأحمال، استهلاك الطاقة.

Related Publications

International Journal Publication

Meriem Gasmi, Mohamed Lamine Kerdoudi, Abdelmalik Bachir. *Load-balanced attribute-based outsourced encryption for constrained IoT devices*. Accepted in Computers and Electrical Engineering (July 2024).

National Conference Publication

Meriem Gasmi, Mohamed Lamine Kerdoudi, Abdelmalik Bachir.*Load-Balanced Lightweight Attribute-Based Encryption for the IoTs*. Accepted in Workshop on Internet of Things Security. Mohamed Boudiaf University of M'sila, Algeria (November 30th, 2021).

Talks and Presentations

Meriem Gasmi, Abdelmalik Bachir. *Green Security In IoT for Healthcare Applications*. A poster presented on SaCoNet Autumn School, El-Oued University, Algeria (October 27-28 2018).
Meriem Gasmi, Abdelmalik Bachir. *Towards Green Security for E-health Applications in the Internet of Things*. Paper presented on SCAP'19 Workshop, Biskra University, Algeria (December 2019).

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

General Introduction

1.1 Context

The Internet of Things (IoT) is an emerging technology that allows a wide range of objects, known as "things," to communicate via wireless connections and connect to the global Internet. This technology supports a broad array of services in various sectors, including industry, agriculture, education, healthcare, and more. It refers to a network of everyday items and gadgets equipped with sensors, actuators, and communication capabilities to collect, share, and analyze data autonomously to make decisions and take actions without the need for human involvement. The IoT has transformed a variety of industries: smart homes use technology such as thermostats, lights, and security cameras to increase comfort and energy efficiency; wearable technology and remote monitoring benefit healthcare; predictive maintenance and supply chain optimization transform industries; urban services are improved in smart cities; crop monitoring devices advance agriculture; and connected vehicles enhance transportation [13].

However, IoT has encountered numerous challenges and vulnerabilities due to its widespread use and the extensive data exchanged between devices over the network. The susceptibility of these devices to cyberattacks varies depending on factors such as data type and application domain. For example, in medical applications, attackers could intercept confidential information exchanged between devices, potentially leading to severe consequences, including patient deaths. The higher the need for information privacy, the greater the cybersecurity risks involved. To address these risks, it is crucial to continuously monitor, update, and implement robust security mechanisms such as encryption, authentication, access control, to protect IoT devices from a variety of attacks exploiting security vulnerabilities [14]. This work focuses on healthcare applications, a highly promising field where IoT technologies are expected to bring substantial advancements. Specifically, devices like medical sensors can be embedded in patients' bodies to monitor vital signs such as blood pressure, heart rate, respiration, and electrocardiogram (ECG).

These sensors then transmit the collected data as Electronic Medical Records (EMR) to the cloud for storage, processing, and decision making. This study emphasizes the importance of ensuring both confidentiality and access control of sensitive data, such as EMRs, during transmission and storage in the cloud.

Among the multiple encryption algorithms available to ensure access control and data confidentiality, Attribute-Based Encryption (ABE) [15] is considered the most convenient for achieving fine access control tailored to meet the specific requirements of the medical context. The complexity and sensitivity of the healthcare system necessitate that EMRs be accessible by different members of the medical staff, with accurately defined access rules. For example, a nurse in a cardiology department is restricted to accessing records for patients in that specific department, whereas a doctor, irrespective of their department, may have broader access permissions. Attribute-Based Encryption (ABE) is an advanced form of traditional public key and identity-based encryption methods [16, 17]. In traditional public-key encryption, the source node encrypts messages using the recipient's public key. In identity-based encryption, messages can be encrypted using a string such as the recipient's name or email address. ABE enhances this flexibility by allowing the source node to encrypt messages using multiple attributes, which can then be sent to one or more destinations. Only nodes possessing keys that match the required attributes can decrypt these ABE-encrypted messages.

1.2 Problem Statement

In our thesis, we address the challenges of security vulnerabilities, privacy risks, energy optimization for devices with limited resources, and improving execution efficiency to enhance overall system performance [18]. As these constrained devices often exchange sensitive information from users or organizations, it becomes crucial to prioritize data protection measures and implement robust encryption protocols to prevent unauthorized access and breaches [19]. Furthermore, incorporating power-saving techniques and efficient resource management strategies can extend the lifespan of these devices, ensuring that they operate optimally without excessive resource consumption or causing disruptions. In this context, CP-ABE (Ciphertext-Policy Attribute-Based Encryption) enables precise access control by embedding authorization directly into the encrypted data, ensuring that only those meeting the policy can decrypt messages. Additionally, medical sensors are designed to be minimally intrusive with limited computation resources, as discussed in [20]. As a result, implementing secure transmission of EMRs to the cloud, without incurring high energy consumption, presents an increasing challenge.

Current CP-ABE solutions are not efficient and suitable for use in IoT environments due to the significant limitations of data producers, making it impossible for them to perform the public-key cryptographic functions required by CP-ABE [21]. However, sev-

eral techniques, such as task offloading, cloud computing, and distributed computing, have been proposed to enhance computing efficiency. These approaches focus on decreasing execution time, overhead, and energy consumption, as noted in [6,11,22,23]. Task offloading involves transferring specific tasks from resource-constrained devices to a more powerful server, thereby reducing the device's burden and improving performance. As discussed in [3,7], cloud computing offers remote storage and processing capabilities, reducing dependence on local resources [4]. Distributed computing, on the other hand, divides tasks into smaller sub-tasks that can be processed concurrently by multiple devices, improving task execution efficiency and resource utilization.

To address these challenges, our aim is to answer the following research question:

- **RQ**: *How can CP-ABE be tailored to achieve fine-grained access control in IoT environments while ensuring secure data transmission, energy efficiency, and sustained performance for resource-constrained devices?*

In order to answer this question, we divided and defined the problem of implementing CP-ABE in IoT for fine-grained access control through the following sub-questions:

- **RQ1**: *How can computation be effectively distributed from different sensors to a set of assistant nodes with different capacities to reduce execution time and improve overall system efficiency?*

  In this research question, the focus will be on optimizing the distribution of computation from sensors to assistant nodes based on their capacities. The goal is to minimize execution time and maximize system efficiency. This will be achieved through the development of a novel algorithm that dynamically allocates tasks based on each node's computational capabilities.

- **RQ2**: *How to deal with data with varying levels of priority where urgent data should be sent first?*

  In this research question, we focus on prioritizing urgent messages in the network by implementing a scheduling mechanism that gives precedence to critical data transmissions. In addition, we must ensure that the proposed strategy prevents urgent data from being delayed or dropped due to network congestion. We will also assess the effectiveness of our approach to evaluate its practicality and reliability, in enhancing communication efficiency and response times in critical situations.

- **RQ3**:*How do we effectively secure constrained IoT device data using CP-ABE without compromising energy efficiency?*

  CP-ABE is a powerful cryptographic technique that enables fine-grained access control for IoT device data. However, its computational demands can significantly affect the energy efficiency of constrained IoT devices. Given the importance of conserving

energy in these resource-limited devices to ensure longer lifespans, this research seeks to find a balance between security and energy efficiency when applying CP-ABE. Specifically, the focus is on optimizing the computational overhead of CP-ABE to enhance energy efficiency while ensuring that data security remains uncompromised.

- **RQ4**: *How to evaluate the performance of our solution in terms of execution time and energy consumption?*

  To address this research question, we have to conduct a series of experiments to measure the execution time and energy consumption of the proposed solution. This will involve implementing the optimized CP-ABE simulator and running various security and energy efficiency tests. By collecting data on the performance metrics, we will be able to assess the effectiveness of our approach in balancing security and energy efficiency in IoT devices. Additionally, we will compare the results against existing solutions to validate the improvements achieved through our research.

In the next section, we will outline the contributions of our proposed approach to IoT security and provide a detailed explanation of how each research question was addressed.

1.3 Contributions

Our objective is to answer the research questions in the above section by proposing a lightweight security solution that balances the need for strong access control measures with the constraints of energy efficiency and device performance.

To overcome the challenges of security and energy efficiency, we propose a lightweight CP-ABE solution suitable for the IoT. The solution involves outsourcing complex encryption tasks to multiple assistant nodes in the network through task offloading and distributed computing.

1. *Load-Balanced Attribute-Based Outsourced Encryption for Constrained IoT Devices*

   Our first contribution is the reduction of encryption operation time by optimizing the distribution of outsourced workloads, which enhances the overall efficiency of the encryption process. We achieve this by assigning workloads to nodes based on their computational capacity, ensuring that more powerful nodes handle larger tasks while less capable nodes receive appropriately sized workloads. This load-balancing approach not only improves encryption execution time, but also ensures efficient utilization of network resources. Moreover, by effectively managing the workload distribution, we can prevent overloading any single node, leading to a more stable and reliable encryption process. To demonstrate the practicality of our solution, we apply it to healthcare applications, where efficient encryption of sensitive data such as patient records is crucial for ensuring privacy and security.

2. *Prioritization of Collected Data in IoT Systems*

   To address data prioritization, our solution takes into consideration the case of heterogeneous sensors with varying levels of priority, ensuring that urgent data is encrypted first. We categorize messages into two types: urgent (high priority) and regular (low priority). To manage this, we introduce a new entity called "the coordinator," which uses different methods to assign schedules for each type of message. For urgent messages, the coordinator enables the sensor to distribute the workload across all available resources, ensuring rapid processing. For regular messages, the coordinator schedules tasks based on resource availability, ensuring efficient allocation while maintaining the quick handling of urgent messages. This prioritization approach is particularly vital in healthcare applications, where the timely processing of critical data, such as emergency alerts or vital patient signs (e.g., heart rate, blood pressure, etc.), can directly impact patient outcomes. By implementing our solution in a healthcare context, we address the high-stakes nature of such environments, ensuring that life-critical information is handled with the urgency it requires while maintaining overall system efficiency for non-urgent tasks.

3. *Efficient Attribute-Based Encryption for IoT: Bridging Security and Energy Efficiency*

   To address the third research question, our third contribution focuses on reducing the energy consumption of CP-ABE in constrained IoT devices. Our solution leverages task offloading and dynamic switching between sleep and active states to achieve this. By offloading computationally intensive tasks, the workload on constrained nodes is minimized, thereby lowering energy consumption. Furthermore, energy usage in the sleep state is significantly lower compared to the active state, further conserving power. Our approach also proposed distributed computing, which reduces communication rates, contributing to further energy savings for IoT nodes.

4. *Experimentation and Evaluation*

   To address the last research question, we have developed our own attribute-based encryption simulator, known as Load Balancing ABE (LB-ABE-Sim), and it is freely available for use. We have used this simulator to evaluate our proposals through a series of experiments, providing valuable insights into their effectiveness. The results demonstrate that our approach improves computation time, reduces energy consumption, maintains data confidentiality, and offers detailed access control compared to current outsourcing ABE solutions.

1.4 Outline

The remainder of this dissertation is organized as follows:

In Chapter 1.4, we provide an overview of the Internet of Things and attribute-based encryption, beginning with their definitions and basic concepts. We then explore various applications and challenges in the field, followed by a discussion on attribute-based encryption schemes, including their properties and types.

In Chapter 2.6, we examine security in the Internet of Things, addressing various standards and lightweight security objectives. Additionally, we explore access control mechanisms, discussing their types and suitability for IoT devices.

In Chapter 3.4, we examine existing research by conducting a thorough review of the literature and findings in the field of attribute-based encryption in the IoT.

In Chapter 4.6, we discuss load-balanced attribute-based outsourced encryption for constrained IoT devices, a method that distributes encryption tasks among multiple IoT devices to enhance both efficiency and security. This contribution introduces a novel approach to the distribution of encryption tasks among multiple IoT devices, aiming to improve efficiency and security through the outsourcing of heavy encryption processes. In addition, we present the Efficient Energy Consumption Solution for ABE in IoTs, a method designed to optimize energy usage in attribute-based encryption (ABE) schemes for IoT devices.

In Chapter 5.7, we present the experimental results and compare our proposed solution with existing methods, highlighting its effectiveness in terms of execution time, energy consumption, and security.

In Chapter 6.9, we conclude the thesis by summarizing our contributions and discussing potential directions for future research.

Chapter 2

Internet of Things and Attribute-Based Encryption

2.1 Introduction

The Internet of Things (IoT) is considered one of the most dominant technologies that will have a great impact on the way we live [24]. It has the potential to revolutionize various industries and enhance efficiency in our daily lives. It enables us to connect billions of objects and communicate with each other in real-time, creating a vast network of interconnected devices and systems. Through the integration of these objects into complex systems, autonomous systems will be able to communicate with each other and monitor their physical surroundings through the use of actuators (devices that initiate action) and sensors (devices that detect changes). It has gained considerable attention in different application fields [25] due to its ability to process large amounts of data quickly and accurately. In healthcare, IoT enables real-time patient monitoring, doctor tracking systems and vital signs tracking [26]; in agriculture, it supports environmental monitoring, industrial plant management, and cattle tracking [27, 28]; in transportation, it enhances traffic management and real-time safety systems in smart vehicles; in public safety and military applications, it aids in logistics, surveillance, and personnel monitoring; in environmental monitoring, it is used for air quality, noise and waterway assessments; and in sectors like smart cities and education, it drives innovation. The broad applicability of IoT has led to its widespread adoption across industries.

The communicated data are often sensitive, confidential, or related to the user's private life. Therefore, it is important to ensure that this data is protected and secure. Compromising this exchanged data may have a big impact on user's life. Security and access control are essential in the IoT domain to avoid access to private information by third parties. A strict access policy needs to be implemented to safeguard sensitive data and maintain data integrity. This can be achieved through encryption, authentication,

and authorization mechanisms.

2.2 Internet of Things Basics

2.2.1 Definition of IoT

The Internet of Things (IoT) is a term that refers to interconnected devices in the form of a network that communicate with each other to share collected information. These devices are connected to the Internet, enabling them to exchange data with other systems [29]. In other words, they are heterogeneous devices capable of collecting, processing, and communicating data and providing various application services [30]. IoT offers benefits such as real-time monitoring, predictive maintenance, improved decision making based on data analytics, and enhanced automation of processes.

The Internet of Things seeks to enable objects to connect seamlessly, anytime and anywhere, at any time, with anything, and with any person, preferably via any path, network, or service [1]. Figure 2.1 represents the six types of IoT.

Figure 2.1: IoT Definition [1].

2.2.2 Internet of Things Applications

With the extensive proliferation of IoT devices, they are present across all domains. IoT is applied in numerous fields to enhance efficiency and simplify our lives, making them more convenient and comfortable [31–33] (see Figure 2.2):

1. *Smart Transportation*

   Smart transportation systems including sophisticated sensors, computers, electronics, and communication technologies are essential to enhance transportation system safety and profitability [34]. These systems combine IoT systems with ICT to save travel time and fuel consumption. Four key concepts scalability, integration, security, and responsiveness are given high priority in the adoption of this technology.

2. *Smart Medical and Healthcare*

   The smart medical healthcare applications [35, 36] combine smart technology, data analytics, and connected devices to improve patient outcomes, personalize treatments, improve overall healthcare efficiency, and increase availability. Examples of such applications include *real-time health monitoring (RTHM), patient monitoring, doctor tracking, early diagnostics, crop health monitoring, personnel tracking, and predictive expertise information to assist doctors and practitioners*. The private data exchanged needs to be kept secure and encrypted to protect patient privacy and confidentiality and prevent unauthorized access. Additionally, these smart medical healthcare applications offer real-time insights, alerts, and recommendations to healthcare professionals for better decision-making and patient care.

3. *Smart Home/building*

   The Smart Home and Smart Building concept is a broad initiative to integrate advanced technologies [37,38], data analytics, and communications systems to enhance efficiency, sustainability, and quality of life within residential, commercial, and urban environments. For instance, smart buildings equipped with IoT-enabled sensors can monitor energy usage in real time, automatically adjusting lighting, heating, and cooling systems to optimize efficiency.

4. *Smart Agriculture and Breeding*

   Intelligent farming systems [39] enhance agricultural productivity by providing agronomists with a better understanding of plant growth models, enabling efficient practices, and mitigating the risks of inappropriate farming conditions.

5. *Environmental Monitoring*

   In order to monitor and control environmental conditions, smart environmental monitoring makes use of cutting-edge technologies, sensors, and data analytics [40]. This allows for real-time data collection, analysis, and well-informed decision-making to address environmental issues and improve sustainability.

6. *Smart Wearable*

   Smart wearables are electronic devices worn on the body that provide real-time data and functionalities to users through embedded sensors and connectivity features [41–43]. They enhance various aspects of life, including health, fitness, communication, and entertainment. This technology offers continuous and precise monitoring in real-time, meeting a variety of needs and preferences.

7. *Smart Education*

   Smart education represents the new era of technologies in this domain [44, 45]. It aims to improve the quality of education, making it more interactive and collaborative in all fields: administration, classroom, assessment, and pedagogy. Smart teaching techniques provide noticeable benefits for learners, educators, administrators, and parents. They motivate learners and enhance their engagement, meeting evolving needs by making education more efficient and effective.

8. *Smart grid*

   A network of computers, software, and communication tools for energy management, leverages IoT technology to reduce economic losses and enhance the efficiency of power transmission [46]. Smart grid technology is essential for achieving a sustainable and secure energy future, offering optimized energy solutions with enhanced security, operational insights, flexible resource management, and improved maintenance forecasting. This technology is transforming how energy is generated, stored, and utilized, creating a new smart energy ecosystem where the traditional electric grid depends on the one-way flow of electricity.

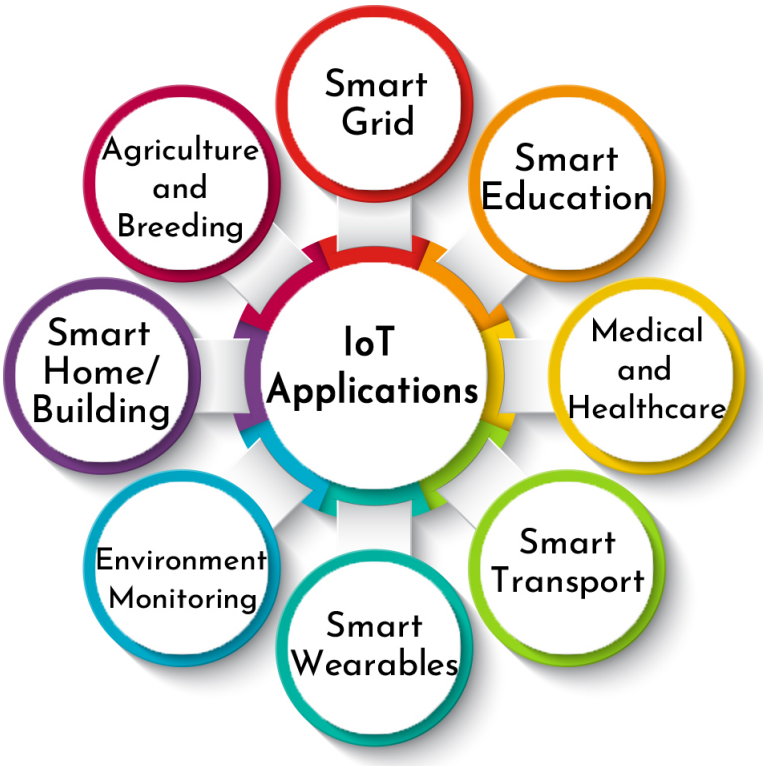


Figure 2.2: IoT Applications.

2.3 Challenges and Limitations in IoT

The Internet of Things (IoT) is a transformative technology reshaping our world. However, like any rapidly evolving innovation, it faces several challenges and limitations [33,47–50]. This section will address the main challenges facing the IoT (see Figure 2.3). Understanding these issues is essential to the successful development and deployment of efficient and secure IoT systems.

1. *Resource Constraints*

   This thesis addresses the critical challenge of resource constraints in IoT systems. IoT devices are inherently limited in processing power, bandwidth, storage capacity, and energy, posing significant obstacles to implementing robust security mechanisms [48]. These limitations make it challenging to deploy complex encryption algorithms and advanced security protocols. To overcome this hurdle, researchers and developers are focusing on designing lightweight security solutions that offer adequate protection while minimizing resource consumption. By tailoring these solutions to the unique constraints of IoT devices, it is possible to enhance security without compromising efficiency or functionality.

2. *Security and Privacy*

   Maintaining privacy and security in IoT applications is a critical challenge [51, 52]. IoT devices record huge amounts of data about people's daily lives , which, when analyzed, can reveal sensitive personal information. Moreover, the interconnected nature of IoT systems amplifies the risks; a security breach in one device can propagate across the network, leading to a domino effect of vulnerabilities. So security is a big concern, and minor alterations to IoT data can result to serious situations. To address these concerns, the implementation of strong encryption protocols and secure communication channels is essential to protect the integrity and confidentiality of data transmitted between devices. Consequently, it is crucial for manufacturers and developers to prioritize security measures in the design and implementation of IoT systems to mitigate potential risks and protect user privacy effectively.

3. *Access Control*

   Ensuring that only authorized users can access and control IoT devices is a critical challenge in IoT security [53]. Managing access control becomes increasingly complex due to the vast number of interconnected devices within IoT networks. Additionally, the diversity of devices and communication protocols in the IoT further complicates this challenge, highlighting the necessity of a unified access control approach that accommodates varying device types and standards. By implementing a robust and flexible access control strategy, organizations can better protect their IoT devices and networks from unauthorized access and potential cyber threats, ensuring both security and operational integrity.

   The challenges outlined above are the central focus of our thesis. We aimed to develop a security and access control solution specifically tailored to resource-constrained IoT devices. Our lightweight solution is designed to overcome the limitations of energy and processing power in these devices, ensuring the implementation of effective and reliable security measures without compromising performance.

4. *Heterogeneity*

   The heterogeneity challenge in IoT refers to the wide variety of devices and communication protocols, and technologies involved, which can lead to security vulnerabilities and disrupt the seamless functioning of IoT networks [47]. This diversity can cause communication issues and require thorough planning for deploying and managing different hardware and software. Overcoming these obstacle is crucial for unlocking the full potential of IoT technology, as it ensures smooth and secure data exchange and system integration.

5. *Interoperability*

   Another significant challenge facing the Internet of Things is achieving interoperability among heterogeneous devices [54]. This challenge arises due to the diverse protocols, standards, and communication languages used by different IoT devices. Without effective interoperability, these devices would not be able to communicate with each other effectively, reducing the potential for seamless integration and collaboration. Device manufacturers as well as application developers need to address this problem. To satisfy customer needs, every IoT service must be built with interoperability issues in mind. Furthermore, IoT developers and manufacturers must prioritize the development of universal standards and protocols that enable interoperability among all IoT devices.

6. *Standardization*

   One of the major challenges in IoT is the lack of standardization across devices and protocols. This makes it difficult for different devices to communicate with each other seamlessly, leading to compatibility issues and security vulnerabilities. Without a universal standard in place, IoT adoption and integration become more complex and costly for businesses and consumers alike. As the IoT ecosystem continues to grow, the need for standardized protocols and communication methods becomes increasingly urgent to ensure the smooth and secure operation of connected devices.

   Standardization is needed in areas such as ensuring service quality, payment methods, handling operations, software applications, and hardware connections to guarantee compatibility and effectiveness.

7. *Integration of IoT with subsystems*

   Integrating IoT with various systems is challenging due to the diverse technologies these systems often use, which may not easily interoperate. This can create issues when devices attempt to communicate or share information. Additionally, ensuring that all devices connect to a single network and operate coherently can be difficult. However, with proper planning and implementation, integrating IoT with these systems can enhance efficiency and automation in various fields. Overall, integrating IoT with other systems presents challenges such as technical issues, planning difficulties, and security concerns. However, effectively addressing these can result in significant benefits for the utilization of connected devices.

8. *Scalability*

   Scalability in the IoT refers to the ability to add devices, services, and functions without affecting the system's performance [55]. It requires designing for extensible services, operations, and interoperability. Scalable mechanisms should support new

device registration, lookup, and discovery. This allows for seamless integration of new devices into the system without causing disruptions. Scalability also involves ensuring that the system can handle the increased data and network traffic that comes with adding more devices. By prioritizing scalability during the design phase, IoT systems can easily adapt and grow as needed without sacrificing performance.

9. *Availability*

   Achieving service availability for users at any time and place presents a challenge in the area of IoT, both in terms of hardware and software [55]. This involves ensuring that devices and software are compatible with IoT functions and protocols. Research is conducted to assess the availability of IoT applications during the design phase, aiming to enhance their effectiveness. Availability in IoT is crucial at both the software level, which guarantees user access to services whenever needed, and the hardware level, which supplies the essential data for application services. Given that these applications are fully dependent on sensor data, the importance of both hardware and software availability becomes essential.

10. *Reliability*

    Reliability entails the capability of software, hardware, and network components to execute their required functions under diverse conditions and at designated times, necessitating that reliability considerations be integrated into every layer of the IoT architecture [55].

    Ensuring the proper functioning of the system is paramount for the successful delivery of services. Availability and reliability are intertwined, guaranteeing that the system remains consistently accessible and trustworthy, which is vital for service delivery excellence. In applications like emergency response, where efficient communication is key, the importance of reliability cannot be overstated it is fundamental at every tier of the IoT infrastructure.

11. *Data analytics*

    One of the main challenges of analyzing data in the IoT is the huge amount of data that is generated by connected devices [47]. This massive volume of data can easily overwhelm traditional data analysis tools and techniques, requiring specialized analytics software and algorithms to process and interpret it effectively. Moreover, the high velocity at which data is being generated in real-time adds another layer of complexity to the analysis process. As more and more devices become connected to the IoT, finding efficient ways to extract meaningful insights from this data will be crucial for organizations to stay competitive and innovative.

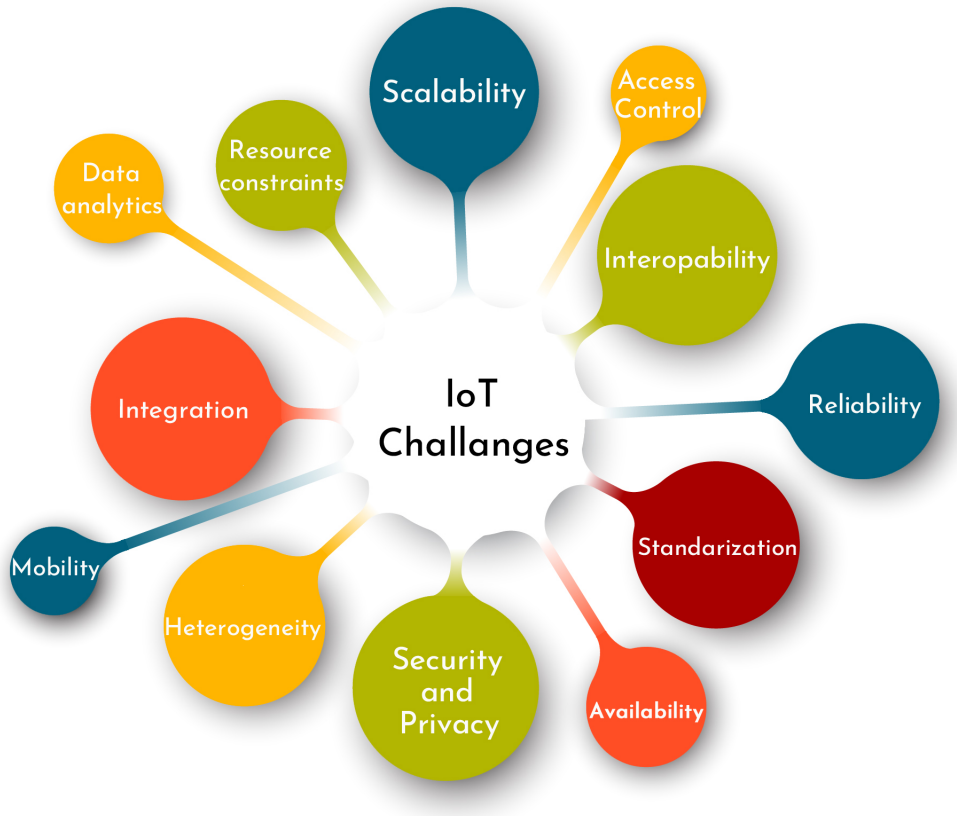


Figure 2.3: IoT Challenges and Limitations.

12. *Mobility*

    The IoT revolution presents a mobility challenge, as services must be continuously delivered to mobile users. Mobility issues arise from device transfer, data accessibility, and resource unavailability [55]. IoT devices, such as smartphones, laptops, and autonomous vehicles, require effective mobility management. However, existing protocols like mobile IPv4 are difficult to incorporate into the IoT due to processing time and energy consumption. To address these limitations, open-source operating systems can help in developing platform-independent applications and cross-interoperability solutions.

2.4 Attribute Based Encryption

In this section, we introduce a public-key encryption system known as attribute-based encryption (ABE), first proposed by Goyal et al. [15]. ABE is designed to provide robust and fine-grained access control based encryption and decryption processes on a set of user attributes.

This powerful encryption scheme is mostly used in cloud computing systems [56] to secure stored data and in public/subscribe architectures, as it ensures keeping confidential

the encrypted data, even when handled by untrusted parties.

ABE comes in two primary variants:

1. **Ciphertext-policy ABE (CP-ABE)** [57]: In this model, the ciphertext is associated with an access structure, while the private key is associated with a set of attributes.

2. **Key-policy ABE (KP-ABE)** [15]: In this model, the ciphertext is associated with specific attributes, while the private key is associated with an access structure.

In addition to its efficiency in ensuring precise access control, ABE suffers from several drawbacks, such as its complexity and high computation overhead, which make the implementation of such a scheme in resource-constrained devices such as IoT difficult. However, the adoption of such a scheme is necessary to satisfy the security and access control requirements of IoT environments.

2.4.1 Definition of ABE

Attribute-based encryption is a form of asymmetric cryptography designed to provide fine-grained access control [58]. Based on the use of attributes and access policies to determine access rights, ABE ensures that encrypted files can only be decrypted by authorized users.

Attribute-based encryption consists of four main primitives: `Setup`, `Encrypt`, `key Generation`, and `Decrypt`. The `Setup` and `key Generation` algorithms are managed by an attribute Authority (AA).

The **Attribute Authority** (AA) is responsible for generating public keys (via the `Setup` algorithm) and provides users with secret keys based on their attributes (through the `key Generation` algorithm). It also manages the universe of attributes.

The **data owner** (DA) defines the access policy for their data and executes the `Encryption` algorithm to secure the data before uploading it to the cloud for storage or public sharing. The obtained ciphertexts are labeled with sets of attributes, while private keys are tied with access structures that determine which ciphertexts a user can decrypt [15].

2.4.2 Preliminaries

1. **Bilinear maps**

   Consider two multiplicative cyclic groups, G_0 and G_1, both having a prime order p. Let g be a generator of G_0, and define e as a bilinear map $e : G_0 * G_0 -> G_1$. This bilinear map e adheres to the following properties:

   - Bilinearity: For all $u, v \in G_0$ and $a, b \in Z_p$, it holds that $e(u^a, v^b) = e(u, v)^{ab}$.

- Non-degeneracy: The map satisfies $e(g, g) \neq 1$.
- Efficient Computability: G_0 is called a bilinear group if both the group operation in G_0 and the bilinear map $e : G_0 * G_0 -> G_1$ can be computed efficiently. Additionally, the map e is symmetric, as $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

2. **Universal attributes set (U):**

   A set of all attributes that define user properties, data properties, and environment properties.

3. **Access tree** T

   - **Definition** : Let T denote a tree that represents an access structure. Each non-leaf node in the tree corresponds to a threshold gate, which is defined by its child nodes and a specific threshold value. If num_x represents the number of child nodes of a parent node x, and k_x is its threshold value, then $0 < k_x \leq num_x$. A threshold gate acts as an OR gate when $k_x = 1$, and as an AND gate when $k_x = num_x$. Each leaf node x in the tree is associated with an attribute and has a threshold value of $k_x = 1$.

     The following functions have been defined to make working with access trees easier:

     - *att(x)*: determines whether x is a leaf node and specifies the attribute associated with the leaf node x in the tree.
     - *index(x)*: indicates the position of the node x among its brothers. The children of a node are numbered sequentially from 1 to num, where num is the total count of nodes sharing the same parent, including x itself.
     - *parent(x)*: returns the parent node of x within the access tree.

   - **Satisfying an access tree** : Consider an access tree T with a root node r. Let T_x represent the sub-tree of T that is rooted at node x. Consequently, the tree T is equivalent to T_r. If a set of attributes γ satisfies the access tree T_x, this expressed as $T_x(\gamma) = 1$. The computation of $T_x(\gamma)$ is done recursively as follows:

     For a non-leaf node x, evaluate $T_{x\prime}(\gamma)$ for all children $x\prime$ of node x. The value $T_x(\gamma)$ is 1 if and only if at least k_x children return 1. For a leaf node x, $T_x(\gamma)$ is 1 if and only if $att(x) \in \gamma$.

The access tree shown in Figure 2.4 is created based on the logical expression ((Speciality = Physician AND Working Place = Hospital AND Department = Pediatrics) OR ((Department = Emergency Medicine OR Department = Pediatrics) AND (Speciality = Nurse)) OR (Speciality = Physician OR Division = Pediatric Cardiology)) that specifies

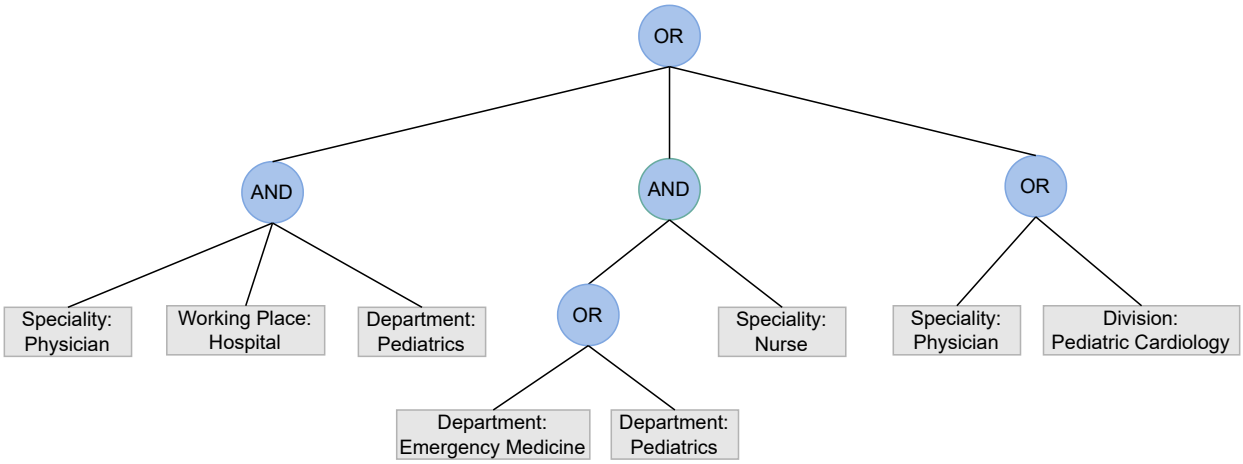
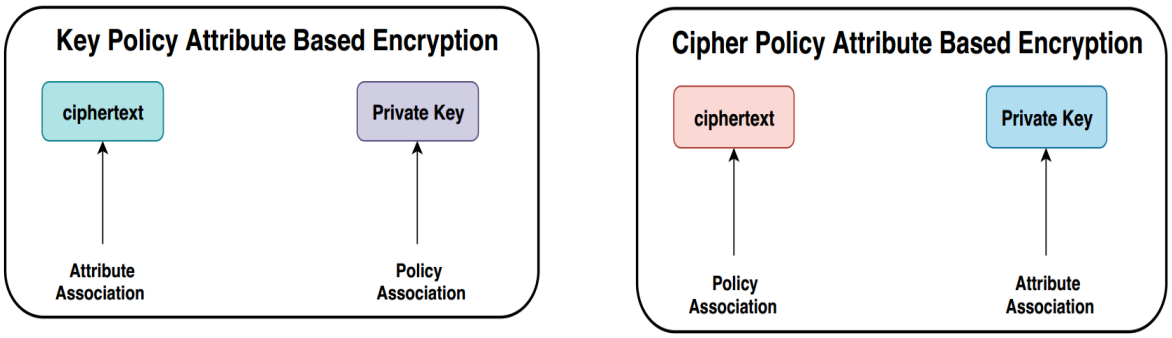


Figure 2.4: Access Tree [2].

conditions for various roles and departments. This logical expression determines which users are authorized to view patient records within the system. All physicians working in a hospital in the pediatric department or whose division is pediatric cardiology can access the data. The same access is granted to all nurses working in the cardiology department or emergency medicine.

2.4.3 ABE Schemes

The two primary Attribute-Based Encryption (ABE) schemes are Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [57] and Key-Policy Attribute-Based Encryption (KP-ABE) [15], as illustrated in Figure 2.5.



(a) KP-ABE **(b)** CP-ABE

Figure 2.5: ABE Schemes.

2.4.3.1 CP-ABE

The CP-ABE scheme is built on four core algorithms: (1) `Setup`, (2) `Encrypt`, (3) `Key Generation`, and (4) `Decrypt`.

- **Setup:**

  This step defines the set of universal attributes and generates the public key (PK) used for the encryption and decryption steps:

$$PK = (G_0, g, h = g^{\beta}, f = g^{1/\beta}, e(g,g)^{\alpha}) \tag{2.1}$$

  and the master key (MK) is used during the key generation process to create the user's secret key.

$$MK = (\beta, g^{\alpha}) \tag{2.2}$$

  where G_0 is a bilinear group of prime order p with generator g, and α, $\beta \in Z_p$ two random exponents.

- **Encrypt** (T, PK, M)**:**

  The encryption process takes as input the access policy (T) and the public key to encrypt the message M. The algorithm encrypts M using the access structure T by assigning a polynomial q_x to each node x in the tree T, including the leaf nodes. These polynomials are defined in a top-down manner, starting from the root node R. For each node x, the degree d_x of the polynomial q_x is set to be one less than the threshold value k_x of the node, i.e., $d_x = k_x - 1$. At the root node R, a random value $s \in Z_p$ is chosen and assigned as $q_R(0) = s$. Then, d_R additional points are randomly selected to fully define the polynomial q_R. For any other node x, the value $q_x(0)$ is determined by the value of its parent's polynomial at the node's index, i.e., $q_x(0) = q_{parent}(x)(index(x))$. The remaining d_x points are randomly chosen to define q_x.

  Let Y represent the set of leaf nodes in T. The ciphertext is constructed by giving the access structure T and computing:

$$\begin{aligned} CT = (T, \tilde{C} = Me(g,g)^{\alpha s}, C = h^s, \\ \forall y \in Y : C_y = g^{q_y(0)}, C_y' = H(att(y))^{q_y(0)}) \end{aligned} \tag{2.3}$$

- **Key Generation** $(MK,\ \gamma)$**:**

  The key generation algorithm is performed by the Attribute Authority (AA). It takes the master key (MK) and a set of attributes γ as input and generates a secret key SK for decryption. This process is repeated for each new user joining the network to compute their respective secret keys.

The algorithm begins by selecting a random value $r \in Z_p$ and another random value $r_j \in Z_p$ for each attribute $j \in \gamma$. The secret key SK is then computed as follows:

$$\begin{aligned} SK = (D = g^{(\alpha+r)/\beta}, \\ \forall j \in \gamma : D_j = g^r H(j)^{r_j}, D'_j = g^{r_j}) \end{aligned} \tag{2.4}$$

- **Decrypt** (SK, CT, PK)**:**

  In this step, the user uses their secret key SK and public key PK to decrypt the ciphertext CT. The secret key is derived during the `Key Generation` step based on the user's attributes. If the user's attributes do not satisfy the access policy in the access tree, they will not be able to decrypt the ciphertext, otherwise, they will retrieve the plaintext M from CT.

  A recursive function, $DecryptNode(CT, SK, x)$, is defined within this algorithm. It takes three inputs: the ciphertext $CT = (T_{cpabe}, \tilde{C}, C, \forall y \in Y_{cpabe} : C_y, C'_y)$, a secret key SK associated with an attribute set γ, and a node x from the access tree T_{cpabe}. The node x must be a leaf node and correspond to an attribute in the set γ. For a leaf node x where $i = att(x)$, the function is defined as follows: If $i \in \gamma$, then:

$$\begin{aligned} DecrypteNode(CT, SK, x) &= \frac{e(D_i, C_x)}{e(D'_i, C'_x)} \\ &= e(g, g)^{rq_x(0)} \end{aligned} \tag{2.5}$$

If x is a non-leaf node, the algorithm $DecryptNode(CT, SK, x)$ proceeds as follows:

1. For each child node z of x, the algorithm recursively calls $DecryptNode(CT, SK, z)$ and stores the result as F_z.

2. Let S_x be any subset of child nodes z with size k_x (the threshold of x) such that $F_z \neq \perp$. If no such subset exists, the node is deemed unsatisfied, and the function returns $\perp$.

3. Otherwise, the function computes:

   $F_x = \prod_{z \in S_x} F_z^{\Delta_i, S'_x(0)}$

   After simplification, this results in $F_x = e(g, g)^{rq_x(0)}$. Further details of this process can be found in [57].

   After that, M is obtained in this way :

$$M = \tilde{C}/(e(C, D)/A). \tag{2.6}$$

Where $A = DecryptNode(CT, SK, r) = e(g, g)^{rs}$

2.4.3.2 KP-ABE

The KP-ABE scheme also consists of four fundamental algorithms, similar to those in CP-ABE, but with some differences in the initial variables: (1) `Setup`, (2) `Encrypt`, (3) `Key Generation`, and (4) `Decrypt`.

- **Setup:**

  The setup algorithm, executed by the attribute authority in the initialization phase, generates the Public Key PK and the Master Key MK with only the implicit security parameter as input.

  The setup algorithm determines at the beginning the universe of attributes $U = 1, 2, ..., n$. For each attribute $i \in U$, a random value t_i is selected from $\mathbb{Z}_p$. Additionally, a random number y is chosen from $\mathbb{Z}_p$. The PK and the MK are then generated according to the formulas 2.7 and 2.8, respectively. The PK is made publicly available to all entities within the system, while the MK remains private and confidential.

$$PK = (T_1 = g^{t_1}, ..., T_{|U|} = g^{t_{|U|}}, Y = e(g, g)^y). \tag{2.7}$$

$$MK = (t_1, ..., t_{|U|}, y). \tag{2.8}$$

- **Key Generation(T_{kpabe}, MK):**

  The algorithm takes as input the access tree T_{kpabe} and the master key MK. It generates a key that allows the user to decrypt a message encrypted under a set of attributes γ, if and only if $T_{kpabe(\gamma)} = 1$.

  The algorithm runs as follows: a polynomial q_x is selected for each node x (including the leaf nodes) in the tree T_{kpabe}. These polynomials are determined using a top-down manner, starting from the root node r. For every node x in the tree, the degree d_x of the polynomial q_x is set to be one less than the threshold value k_x of that node ($d_x = k_x - 1$).

  For the root node r, initialize $q_r(0) = y$, and randomly choose d_r additional points to fully define the polynomial q_r. For any other node x, set $q_x(0) = q_{parent(x)}(index(x))$, then randomly select d_x other points to completely define q_x. Once all polynomials are defined, for each leaf node x, we assign the following secret value to the user: Let Y_{kpabe} denote the set of leaf nodes in the access tree T_{kpabe}.

$$\forall x \in Y_{kpabe} : Dx = g^{\frac{q_x(0)}{t_i}}.(\text{Where i} = \text{att(x)}) \tag{2.9}$$

- **Encrypt(PK, M, γ):**

  The algorithm takes as input the public key PK, the message M, and the set of attributes γ.

  First, a random value $s \in \mathbb{Z}_p$ is chosen. The ciphertext is then calculated as follows:

$$CT = (\gamma, CT' = MY^s, \{CT_i = T_i^s\}_{i \in \gamma}) \tag{2.10}$$

- **Decrypt(CT, D):**

  A recursive algorithm $DecryptNode(CT, D, x)$, takes as input:

  - the ciphertext $CT = (\gamma, CT', \forall i \in \gamma : CT_i)$,
  - the private key D (we assume the access tree T is embedded in the private key), and
  - a node x in the tree.

  The algorithm outputs either a group element of $\mathbb{G}_2$ or $\perp$. Let $i = att(x)$.

  If the node x is a leaf node, then:

  - If $i \in \gamma$, compute:

$$DecryptNode(CT, SK_{KPabe}, x) = e(D_x, CT_i) = e(g, g)^{sq_x(0)} \tag{2.11}$$

  - Otherwise, the algorithm returns $\perp$.

  If x is a non-leaf node, the algorithm $DecryptNode(CT, D, x)$ proceeds as follows:

  1. For each child node z of x, call $DecryptNode(CT, D, z)$ and store the output as F_z.
  2. Let S_x be an arbitrary k_x-sized subset of child nodes z such that $F_z \neq \perp$. If no such set exists, the node is not satisfied, and the function returns $\perp$.
  3. Otherwise, compute:

$$F_x = \prod_{z \in S_x} F_z^{\Delta_i, S'_x(0)} \tag{2.12}$$

  After simplification, this yields $e(g, g)^{sq_x(0)}$.

  Finally, at the root node r, the algorithm computes:

  $DecryptNode(CT, D, r) = Y^s$

  This holds if and only if the ciphertext satisfies the tree. Since $CT' = MY^s$, the message M is recovered as follows:

$$M = \frac{CT'}{Y^s} \tag{2.13}$$

This process allows for secure decryption of the ciphertext and retrieval of the original message sent by the sender.

2.5 Challenges in Implementing ABE in IoT Environments

Implementing ABE in the IoT environment poses several challenges and limitations [59], as it contains complex and heavy operations. In addition, the reliance on bilinear maps in encryption and decryption algorithms increases the complexity of computation as the number of attributes increases. In our work, we aim to address two main challenges:

2.5.1 Resource Constrains

IoT devices often face limitations in memory, battery life, and processing capabilities, which present significant challenges when implementing complex ABE schemes.

- Limited Processing Power: ABE requires complex cryptographic operations, which are demanding for IoT devices due to their restricted computational capacities.
- Memory Limitations: ABE schemes are memory-intensive and requiring significant storage for keys, attributes, and ciphertexts. This can be problematic for IoT devices with limited memory resources.
- Energy Consumption: Cryptographic processes require substantial energy; therefore, energy efficiency is vital for battery-powered IoT devices. ABE can consume a lot of energy, which significantly reduces the device's lifespan and efficiency.

2.5.2 Latency and Performance

- Encryption/Decryption Time: The complexity of the mathematical operations involved in encryption and decryption algorithms can lead to significant latency. This poses a challenge for real-time applications, where fast response times are crucial.
- Communication Overhead: ABE schemes include the attribute information in the ciphertext, resulting in larger encrypted messages. This increased message size can lead to higher communication overhead, which is particularly challenging in IoT networks with limited bandwidth.

Many studies propose lightweight ABE solutions to address these challenges, as will be discussed in detail in Chapter 3.4.

2.6 Conclusion

This chapter provided a foundational understanding of the Internet of Things (IoT). We examined the definition of IoT systems, explored their potential applications in various fields, and discussed the challenges and limitations associated with IoT. Additionally, we introduced attribute-based encryption (ABE), including its definition, key components, schemes, and the challenges it faces when applied in IoT environments. Addressing these challenges is crucial to ensure a efficient and secure implementation of ABE in IoT systems. In our thesis, we proposed a novel lightweight attribute-based encryption scheme specifically designed for IoT applications. By considering the resource-constrained nature of IoT devices, our approach optimizes the encryption process to minimize computational overhead, execution time, and energy while maintaining robust security compared to existing methods.

Building upon the challenges discussed in this chapter, the next chapter will focus on key security aspects, including data privacy, authentication mechanisms, and network vulnerabilities in IoT systems.

Chapter 3

Security in the Internet of Things

3.1 Introduction

The Internet of Things has revolutionized our interactions with everyday objects, becoming deeply integrated into our daily lives [60, 61]. However, it suffers from issues related to privacy and security, making it a target for hackers and malicious users. As IoT devices exchange vast amounts of sensitive data, particularly in critical domains such as healthcare, rendering them vulnerable to cyberattacks [62]. Consequently, implementing robust security measures is essential to protect these data from unauthorized access and breaches [63]. From inherent vulnerabilities such as weak encryption in IoT devices to the constant threat of data breaches, it is imperative that individuals and organizations proactively secure their sensitive information [64]. To mitigate cybersecurity risks within the IoT ecosystem, several mechanisms can be employed, including multi-factor authentication, access control, robust encryption protocols, and regular security updates.

This chapter provides an overview of common security issues and threats in IoT. The following section delves into IoT security and lightweight security objectives. Finally, we will examine access control, exploring its various types and highlighting their key differences.

3.2 Security Goals in IoT

Depending on the type of application and the required security level, a security solution may have one or more goals. These goals typically include authentication, confidentiality, user privacy, integrity, and intrusion detection. Ultimately, the main goal of the security solution is to protect IoT devices and data from unauthorized access and malicious attacks, creating a secure and trustworthy environment in which connected devices can interact safely.

The goals are classified into two types based on security requirements: standard se-

curity and lightweight security. Standard security focuses on comprehensive protection measures to ensure robust security across IoT systems, while lightweight security focuses on balancing security with efficiency and performance, making it particularly suitable for resource-constrained IoT devices.

3.2.1 Standard Security Goals

– *Authentication & Authorization*

These two factors are pivotal in ensuring the security of IoT devices. Authentication verifies the identity of users or devices accessing the system, while authorization determines what actions they are allowed to perform. Implementing robust authentication mechanisms, such as multi-factor authentication, can effectively prevent unauthorized access to IoT devices. Likewise, establishing strong authorization rules can restrict the activities permitted for users and devices.

Taking a thorough approach to authentication and authorization is vital for protecting IoT devices from security risks [65, 66].

– *Confidentiality*

Confidentiality means ensuring the security of exchanged data by making its content accessible only to authorized entities while remaining unintelligible to unauthorized devices or users. Confidentiality is essential to maintain the privacy and trust for individuals and organizations. Confidentiality is typically achieved using encryption techniques, such as symmetric keys or public key encryption algorithms. For instance, in the medical domain, patient records are encrypted to protect sensitive information from unauthorized access and ensure privacy.

– *Integrity*

Integrity ensures that data exchanged between a sender and a legitimate receiver remains unchanged and accurate during transmission. It protects against unauthorized alterations that could lead to harmful consequences. By prioritizing integrity as a key security goal in IoT systems, organizations can maintain the trustworthiness of their operations and protect sensitive information from unauthorized modifications. Integrity can be ensured using encryption, digital signatures to verify authenticity, and hashing functions to detect alterations. For example, in the medical domain, ensuring integrity prevents tampering with patient records or treatment plans, maintaining the accuracy and reliability of critical healthcare data.

– *Availability*

Availability ensures that a system remains continuously accessible and operational for legitimate users when needed. In the IoT context, availability is crucial, as many

IoT devices are designed to provide continuous monitoring and control over various systems. Ensuring availability involves implementing robust security measures to prevent disruptions or attacks that may compromise the system functionality. These measures may include system redundancies, frequent security updates, and proactive monitoring to detect unusual activity that could indicate potential threats. The continuous availability of IoT devices is essential for their proper functioning and reliability. For instance, in the medical domain, continued patient monitoring is vital for timely medical interventions and improving patient outcomes [67].

– *Non-repudiation*

Non-repudiation is a critical aspect of IoT security, ensuring that the origin and authenticity of transmitted data cannot be denied by the sender. By implementing non-repudiation mechanisms, organizations can protect against fraudulent activities and resolve disputes with verifiable proof of communication. This security goal helps to build trust among entities and maintain the credibility of the IoT ecosystem. For example, in the sensitive medical domain, where decisions carry significant consequences, non-repudiation can play a vital role. It provides irrefutable evidence of who sent specific data and when, which is invaluable in addressing medical errors or other disputes.

– *Data Freshness*

To mitigate the risk of replay attacks, maintaining data freshness is crucial, as it ensures the legitimacy of received data and thwarts unauthorized playback. This means that the data being collected and transmitted by IoT devices is current and accurate. Achieving this requires implementing mechanisms for real-time data processing and synchronization. Moreover, encryption and authentication protocols play a critical role in ensuring that the transmitted data remains secure and unaltered. By prioritizing data freshness in IoT security measures, organizations can enhance the reliability and trustworthiness of their IoT systems.

– *Privacy Protection*

The growing volume of data generated by the Internet of Things presents a serious privacy challenges. IoT devices, sensors, readers, and applications collect diverse data types from individuals, which may reveal personal information and preferences. The gathered data may include sensitive details such as locations, hobbies, habits, and other personal attributes, necessitating robust privacy protection measures.

– *Access Control*

Access control involves defining and enforcing rules to determine who can access specific resources or perform specific actions in a large IoT network. It is important

to establish clear policies and robust mechanisms for access control in order to ensure the security and integrity of the IoT systems [53]. For example, a medical database implements strict access control measures to preserve patient confidentiality and prevent unauthorized access to sensitive information.

3.2.2 Lightweight Security Goals

Traditional security solutions are unsuitable for IoT devices due to their constrained nature. Lightweight security solutions are designed to provide robust protection while minimizing resource consumption.

The main goal of lightweight security solutions in IoT is to protect devices and data without compromising performance. By employing advanced encryption algorithms, such as AES, and secure communication protocols, like TLS, these solutions aim to prevent unauthorized access and data breaches within IoT networks. Furthermore, lightweight security approaches use optimization techniques and energy-efficient technologies to reduce resource consumption and ensure seamless operation of connected devices [68]. Key goals of lightweight security solutions typically include:

- *Low computational overhead*: Ensure efficient processing of security protocols without significantly impacting device performance.
- *Low power consumption* : Ensuring the device can operate for extended periods without draining the battery, by maximizing its lifespan [69, 70].
- *Low computation time*: Reducing delays in executing tasks, making the device more responsive.
- *Low memory requirements*: Facilitating the running of multiple applications simultaneously.
- *Low storage requirements*: Allowing the device to store more data and files.
- *Low latency*: providing faster response times for a better user experience.
- *Improved security performance*: Ensuring data protection and confidentiality and increasing overall productivity and efficiency.

Energy remains a critical factor in IoT security [71], especially for resource-constrained devices that must perform autonomous operations over extended periods. This is particularly vital in applications involving sensor-based continuous monitoring or in remote or inaccessible locations.

Interruptions caused by excessive energy consumption can be life-threatening in critical scenarios and can significantly disrupt the continuous functionality of IoT systems.

Figure 3.1: Balanced Security Objectives in IoT Environments.

In addition to energy consumption, ensuring data protection, security, and optimal performance is crucial for IoT systems. Achieving a balance between energy efficiency, robust security, and high performance is essential in IoT systems for their overall effectiveness 3.1.

3.3 Access Control in IoT

Access control is essential for managing permissions and securing interactions between organizations, users, and connected IoT devices. By setting specific access rules, organizations can ensure that only authorized individuals or devices can interact with IoT systems. This prevents unauthorized access, mitigates data breaches, and addresses potential security vulnerabilities. In addition, access control enables users to monitor and track access to IoT devices, offering valuable insights into usage patterns and potential security threats [53].

An effective access control system ensures the confidentiality of sensitive information (accessible only to authorized users), the integrity of data (preventing unauthorized modifications), and availability (uninterrupted access for authorized users). A comprehensive access control system comprises three core functions:

- **Authentication**: Verifies user identities.
- **Authorization**: Determines and enforces access rights.
- **Accountability**: Tracks and logs user actions for auditing purposes [72].

3.3.1 Definition of Access Control

Access control refers to the process of managing access requests to resources and data by granting or denying permissions based on established security policies. It plays a crucial role in maintaining the confidentiality, integrity, and availability of information within a system [73].

3.3.2 Access Control Models

Several access control models (see Figure 3.2) have been proposed in the literature to meet the unique security requirements of IoT systems [53, 73–76]. These models aim to provide fine-grained access control while addressing the diverse and dynamic nature of IoT environments. These models possess distinct characteristics, influencing their suitability for specific authorization mechanisms in IoT applications.

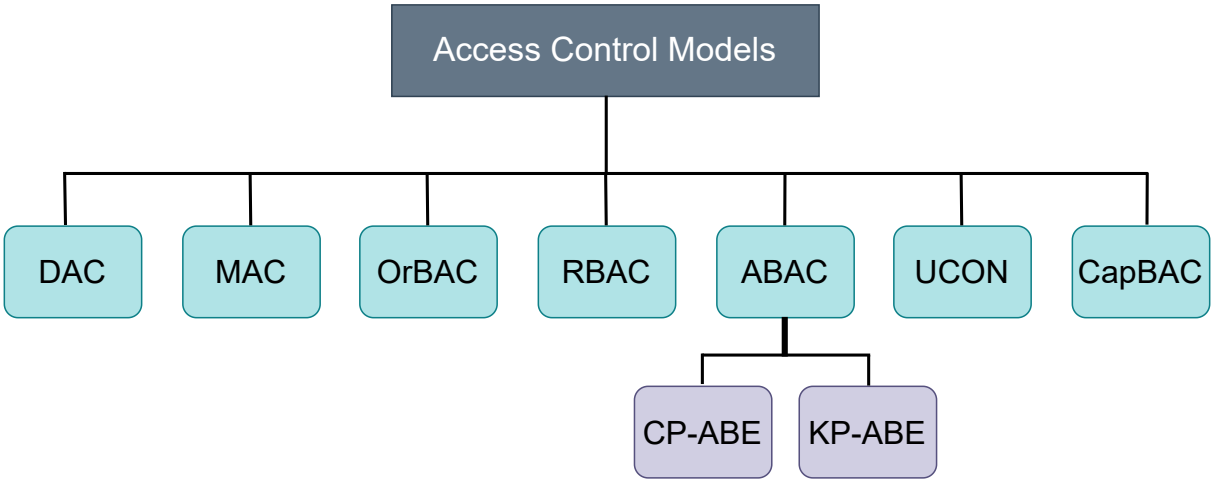


Figure 3.2: Access Control Models.

1. *Discretionary Access Control (DAC)*

   DAC is a user-centric access control model based on ownership, allowing subjects to control access permissions to their objects [77]. It is an identity-based model, assigning access rights based on users' identities and is implemented through mechanisms such as access matrices, authorization table, access control lists (ACLs), and a capability lists. However, DAC is a static model that relies on manual updates of the ACLs by administrator, making it unsuitable for dynamic IoT environments. In IoT, access decisions must be automated and timely updated or revoked to address the continuously changing nature of devices and data.

2. *Mandatory Access Control (MAC)*

   In contrast to Discretionary Access Control (DAC), Mandatory Access Control (MAC) enforces access restrictions based on the sensitivity of the information rather

than the identity of the user or process requesting access [53]. MAC is built around the concept of security classifications (e.g., top-secret, secret, confidential) assigned to both subjects (users or entities with clearance levels) and objects (resources with classification levels). Access is granted based on these security levels, ensuring that only authorized entities can interact with sensitive information. The core components of MAC include a collection of objects, a set of subjects, permissions, and defined security levels, all managed centrally to ensure consistent and secure enforcement of access control policies.

3. *Role-based Access Control (RBAC)*

   Determining roles and permissions inside an organization is essential to RBAC [78]. In RBAC, users are granted permissions based on their assigned roles, which encapsulate the necessary permissions to access specific resources. A single role can be assigned to multiple users, and users can hold multiple roles, such as nurse, doctor, or administrator. A single role may apply to a large number of users, but when new devices with enhanced capabilities are introduced to the network, administrators may need to create additional roles to accommodate them. This can result in role proliferation, making management heavy. Additionally, RBAC's static structure can pose challenges in supporting the adaptability required in dynamic and evolving environments like IoT.

4. *Organization-Based Access Control (OrBAC)*

   OrBAC extends role-based access control by incorporating the concept of "organization" [79]. Similar to RBAC, a position in an organization denotes a job function. By introducing an additional layer of abstraction above RBAC, it seeks to make the formulation of security rules more straightforward. It makes it possible to simulate an access control policy without regard to how it is implemented in the context of an organization. In simpler terms, it provides a method of access control that is more flexible and responsive than RBAC. This level of abstraction, however, can increase the complexity of IoT devices and potentially introduce security vulnerabilities. Moreover, it is not well-suited for the dynamic and heterogeneous nature of IoT environments.

5. *Attribute-based Access Control (ABAC)*

   In contrast to Role-Based Access Control (RABC), ABAC is a type of access control that uses attributes to determine access rights [80,81]. Thus, ABAC has the ability to dynamically assign access rights based on specific attributes. Access rights are limited based on the attributes of subjects (such as role, department, age), objects (such as type, ownership, classification), actions (such as read, write, execute), and the environment (such as time, location, network conditions). The policies and

access requests are defined using attribute names-value pairs, allowing for granular and precise control over access permissions. It is widely regarded as one of the most suitable models for managing access rights in IoT, due to its ability to incorporate additional attributes alongside user roles.

6. *Usage Control (UCON)*

   Like ABAC, UCON permits policy definition based on the attributes of subjects and objects [82]. It provides the same expressiveness as ABAC by using conditions to define access restrictions on the environments. Two further decision qualities that UCON supports are continuity of decision and mutability of attributes. The mutability of attributes allows for the modification of subjects' and objects' attributes during usage, while continuity of decision ensures that permissions are continually verified throughout the entire usage period, not just at the moment of access. Despite these advantages, UCON operates within a centralized architecture and does not account for the delegation property.

7. *Capability-Based Access control (CapBAC)*

   In the CapBAC model, access rights are granted based on the possession of a token of authority, which serves as the basis for accessing objects [53]. The main idea of this access control model is the allocation of access rights to subjects based on their possession of a token, ticket, or key that indicates their authority. This key refers to a special object that has a set of related access privileges. This model's core concept is that access permissions aren't decided by the identity of the requesting entity but rather by matching the relevant capabilities. When CapBAC was first introduced, access rights validations were handled by a central server. The IoT devices in this paradigm guarantee access rights validation. However, in unreliable IoT contexts, these models are unreliable because of the restricted capabilities of IoT and how easily they may be compromised.

Access control models presented earlier are unsuitable for IoT devices due to their limited hardware resources and energy constraints [73, 76]. Instead, there is a need for lightweight access control models specifically tailored for IoT environments. To make these models feasible, several techniques can be employed, such as offloading computation to the cloud, using optimized algorithms designed for resource-constrained devices, maintaining constant ciphertext size [83], adopting online-offline Attribute-Based Encryption (ABE) schemes [84], and leveraging fast cryptographic primitives [85]. These adaptations help ensure the security and efficiency of access control mechanisms in IoT ecosystems while addressing the constraints imposed by the devices' limited resources.

3.4 Conclusion

In this chapter, we explored the standard and lightweight security goals of IoT, emphasizing their importance in enabling secure implementation on resource-constrained devices. We also highlighted the access control mechanisms and models, analyzing their suitability for IoT environments. Based on our analysis, we deduce that Attribute-Based Access Control (ABAC) emerges as the most suitable model for IoT environments due to its flexibility, scalability, and ability to provide fine-grained access control. Its adaptability to the dynamic nature of IoT devices makes it a robust choice. However, some adjustments may be required to fully meet the constraints of resource-constrained IoT devices while preserving their intended functionality.

The next chapter will present a comprehensive literature review of existing lightweight Attribute-Based Encryption (ABE) solutions and the techniques developed to tailor these solutions for resource-constrained IoT environment.

Chapter 4

Literature Review

4.1 Introduction

This chapter reviews the existing literature on attribute-based encryption (ABE) in the IoT environment. ABE is a cryptographic approach where access control is governed by attributes of users and data, rather than their identities. This approach enhances both the flexibility of data sharing and the security of IoT networks by ensuring that only users with specific attributes can access encrypted data. However, the limited processing power, memory capacity, and battery life of IoT devices pose significant challenges to the development of robust attribute-based access control systems. These limitations can affect the performance and efficiency of ABE mechanisms in IoT applications. Consequently, it is imperative to devise innovative solutions to overcome these limitations and guarantee secure data access within IoT environments.

By reviewing previous studies and scientific articles, we aim to provide a comprehensive overview of the current state of knowledge in this field of attribute-based encryption (ABE) within IoT environments. This literature review serves to contextualize our research and identify critical gaps in the existing literature that our study aims to address. We will examine techniques designed to optimize ABE for lightweight and efficient operation, particularly in resource-constrained IoT devices. This analysis will offer valuable insights into the advancements and limitations of current methods, paving the way for a deeper understanding of our contributions, which are presented in the following chapter. Our research strives to build upon and enhance existing solutions, offering practical and effective improvements to the field.

We categorized the literature review into three categories: outsourcing solutions for CP-ABE, load-balancing and priority-based solutions, and energy consumption of ABE in IoT. Each category addresses a specific aspect of the challenges and solutions related to computation outsourcing in cloud computing environments. This classification provides a comprehensive overview of the existing research and highlights advancements and

limitations in each area.

4.2 Outsourcing Solutions for CP-ABE

In this section, we will delve into the various computation outsourcing solutions that can be utilized to enhance the efficiency and suitability of CP-ABE in IoT.

Ciphertext-policy attribute-based encryption (CP-ABE) was proposed in [57] to address the access control issues in cloud computing. The goal is to realize complex access control on encrypted data that ensures the confidentiality of stored data even when the cloud service provider is untrusted. CP-ABE allows data owners to define fine-grained access policies based on user attributes and encrypt data in a way that only users with the required attributes can decrypt it. However, CP-ABE is not designed to run on IoT-constrained devices, such as those commonly used in the medical context and that are typically installed on patients' bodies. These devices typically have limited computational resources and memory, making CP-ABE unsuitable for such contexts. To address this challenge, our lightweight solution reduces the computational burden on these devices by outsourcing heavy cryptographic operations to more powerful external entities while maintaining the security and privacy of the transmitted data.

In [22], the authors propose to solve the problem of using CP-ABE in IoT environments, as it contains complex and powerful encryption and decryption operations. They proposed a computation outsourcing technique where the constrained node offloads the heavy encryption tasks of ABE to multiple assistant nodes (powerful devices) within the network. The authors split and distribute the costly tasks equally among all available assistant nodes. These assistant nodes then perform the heavy encryption tasks and send the results back to the constrained node and the server node for final computations. They demonstrate the efficiency of their solution in terms of computational cost and security. Our contribution enhances their approach by introducing a distinct workload distribution mechanism. Unlike their method, which evenly divides the workload among assistant nodes, potentially leading to performance bottlenecks when nodes have varying processing capabilities, our approach optimizes performance by allocating the workload proportionally to the capacities of each assistant node.

In [3], the authors address the challenge of frequent data re-encryption whenever the access policies change, aiming to reduce the high computational workload for constrained devices. They propose encrypting the data using symmetric encryption and securing the symmetric key by ABE. To offload constrained devices from complex computations, the symmetric and ABE encryption processes are fully outsourced to a gateway. The encrypted data are stored in the cloud for future use by *Healthcare professionals*, as shown in Figure 4.1 that illustrates their system architecture. The authors prove the efficiency of their proposal in terms of eliminating the potential security threats associated with

medical data outsourcing, ensuring data confidentiality and integrity without involving patients or doctors' interventions.

However, their solution assumes complete trust in the gateway, a limitation addressed in our approach. In our solution, we implement a distributed encryption scheme that divides the encryption process across multiple entities to reduce the risk of a single point of failure. This distribution ensures that no single entity has full access to the sensitive data, thereby adding an additional layer of security to the system.

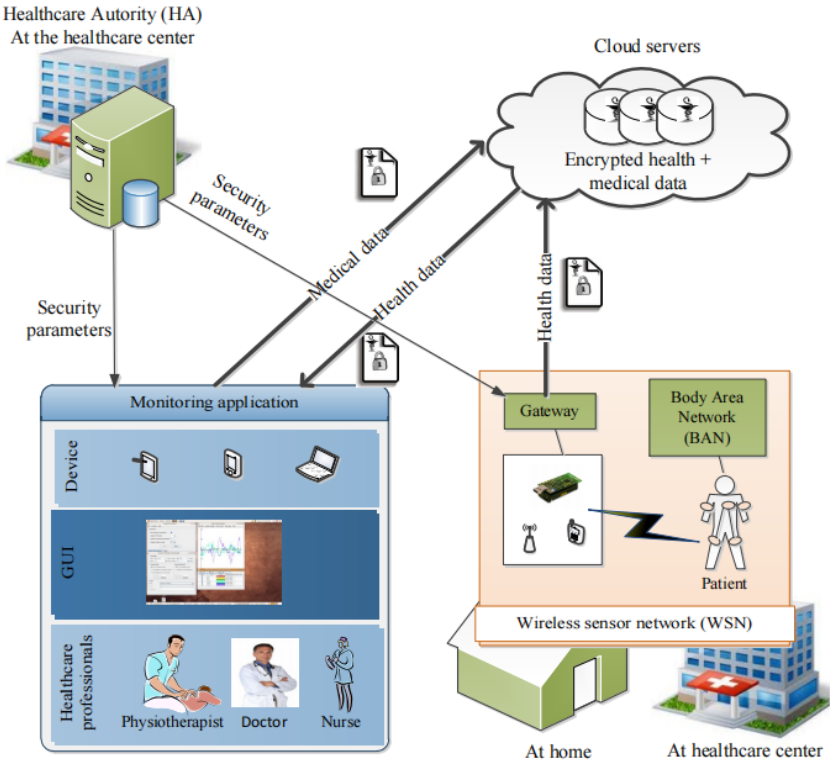


Figure 4.1: Remote monitoring system architecture [3]

Cloud computing is increasingly expanding to mobile environments due to wireless access technologies and the proliferation of mobile intelligent terminals. Although attribute-based encryption (ABE) is commonly used in cloud computing to enforce access control, it results in high computational costs during encryption and decryption, especially as the complexity of access policies increases. The work proposed in [4] addresses these computational limitations in mobile devices by proposing the Verifiable Outsourced Computation Ciphertext-Policy Attribute-Based Encryption Scheme (VOC-CP-ABE). This scheme efficiently delegates computationally intensive tasks to external entities without compromising private information (see Figure 4.2). The scheme ensures both security and verifiability by using two hash functions for verification. Its performance was evaluated and compared with similar solutions, demonstrating its effectiveness. In contrast to their approach, our solution outsources the most encryption calculations to assistant nodes while maintaining a balance between efficiency and data security.

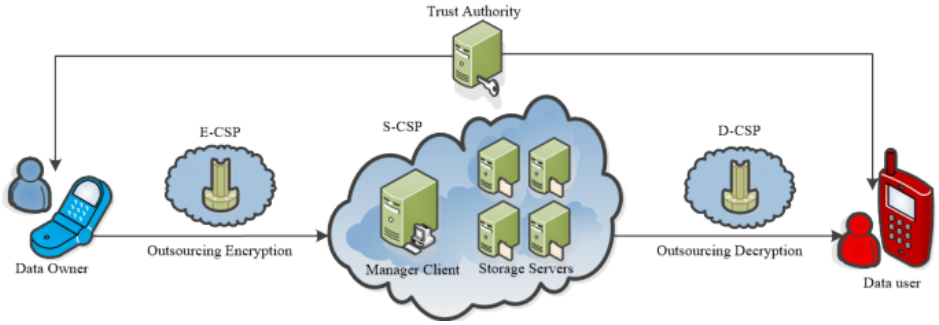


Figure 4.2: Verifiable Outsourced Ciphertext-policy Attribute-based Encryption for Mobile Cloud Computing System Model [4].

In [5], the authors propose an efficient attribute-based encryption (ABE) system that delegates certain encryption and decryption operations to powerful edge nodes, thereby reducing the computational load on resource-constrained devices. In the system model depicted in Figure 4.3, the data owner initially encrypts a portion of the data locally to generate CT_{Local}. This partially encrypted data is then sent to an edge node for the final encryption steps. Likewise, the recipient, known as *Data User*, receives a partially decrypted ciphertext from the edge node before finishing the decryption process locally. While these approaches enable computation outsourcing and alleviate some of the workload on constrained devices, the terminals (such as sensor nodes in our case) are still required to calculate bilinear pairing operations, which can be computationally intensive.

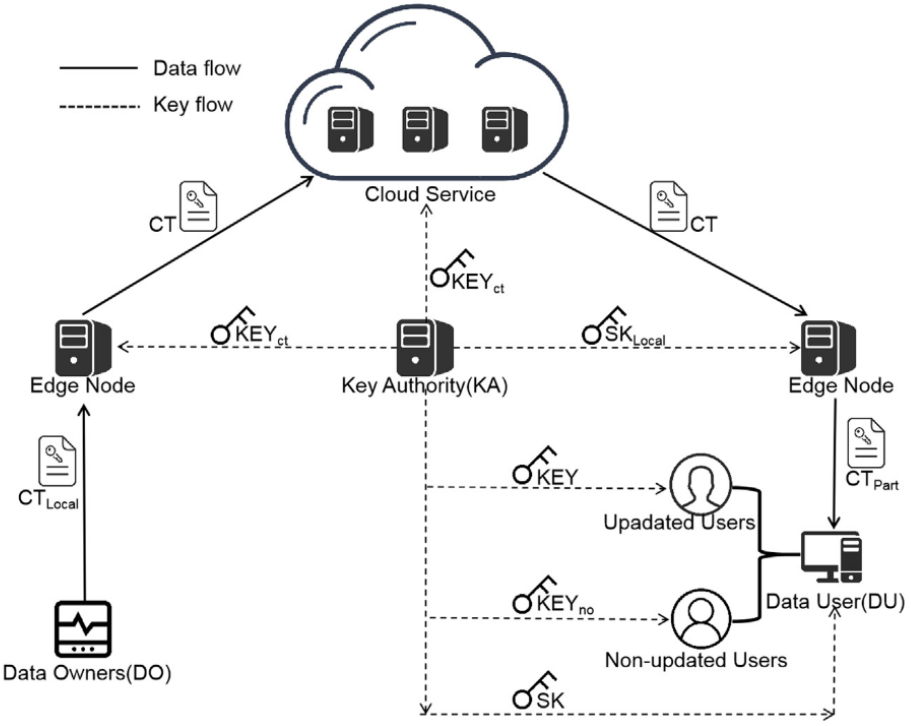


Figure 4.3: An efficient and outsourcing-supported attribute-based access control scheme for edge-enabled smart healthcare System Model [5]

In [86], the authors compared two CP-ABE encryption schemes: the full encryption scheme proposed in [57] and the partial encryption scheme proposed in [87]. Their study evaluated the performance of these schemes under different conditions (constrained node state, server state, file size, policy length) by measuring the execution time and resource consumption (CPU and memory) to determine the most suitable scheme for various scenarios. In their next paper [6], they introduced a smart offloading technique that dynamically switches from full encryption to partial encryption based on available resources and parameters such as the number of attributes, data size, CPU, memory, and battery levels. In both schemes, they add a dummy attribute to the access policy (see Figure 4.4), to be able to use one secret key for decryption, whether full or partial encryption is applied. In the partial encryption scheme, the data is encrypted first with the dummy attribute, and most CP-ABE operations are offloaded to a helper device or proxy server. In contrast, in full encryption, the data is encrypted using the entire access policy. The authors demonstrated the effectiveness of their proposal by evaluating its impact on execution time, CPU utilization, and power consumption. However, their approach still requires the constrained device to execute complex operations in both full and partial encryption schemes, which can be challenging for resource-limited environments.

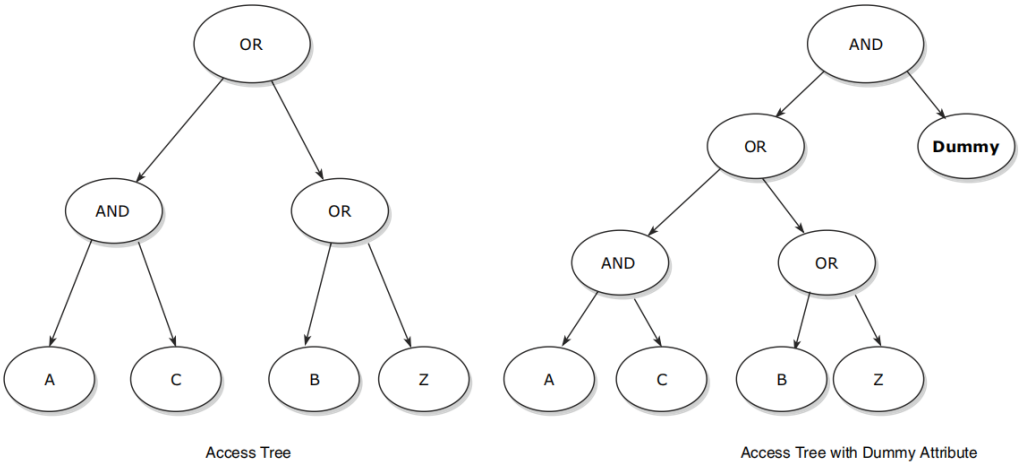


Figure 4.4: Access tree without and with dummy attribute [6].

In [7], the authors propose an encryption outsourcing model in which the data owner outsources computationally intensive CP-ABE encryption operations to a single semi-trusted entity. This approach differs from ours, which employs multiple entities to distribute the computational workload. In their model, the constrained device, referred to as the *DataProducer* (PD), handles a part of the ciphertext generation process locally and delegates the rest to a more capable entity known as the *Delegatee* (DG). The fully processed ciphertext is then either sent to the cloud for storage or sent directly to *DataConsumer* (DC), as shown in Figure 4.5. The authors demonstrate the efficiency of their proposal in terms of energy consumption and execution time compared to existing solutions, using the emulated Wismote sensor platform and a laptop for evaluation. How-

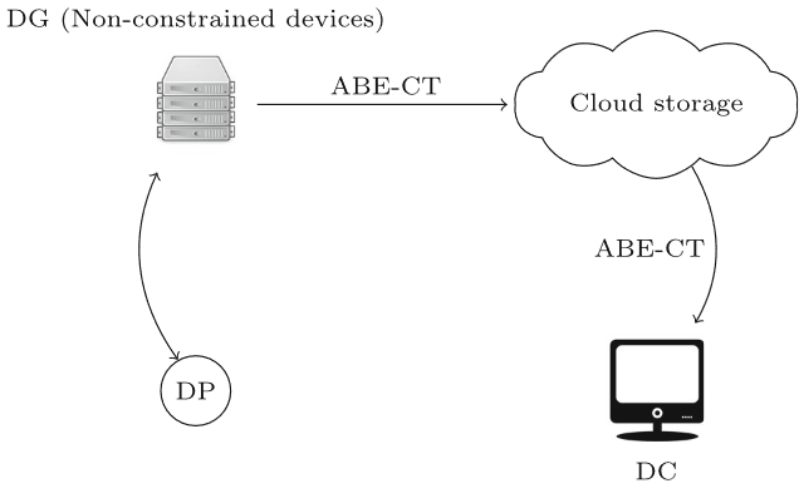
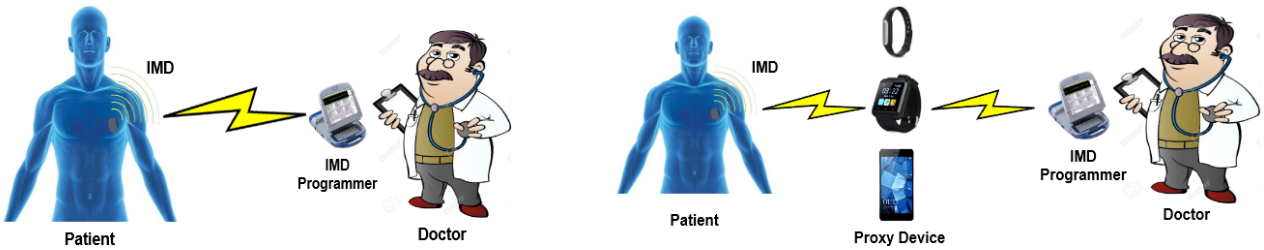


Figure 4.5: Securely Outsourcing the Ciphertext-policy Attribute-based Encryption System Model [7].

ever, their method still requires constrained devices to compute a single exponentiation operation to generate the ciphertext CT, which we identify as a limitation. In contrast, our solution eliminates the need for constrained devices to perform any exponentiation operations.

In [8], the authors propose a lightweight solution tailored for implantable medical devices, which are constrained in terms of computational power, storage, and battery capacity. Their approach delegates resource-intensive operations to a single proxy device such as the patient's smartphone (see Figure 4.6). While effective, we identify this reliance on a single proxy device as a limitation due to the potential lack of fault tolerance. In contrast, our solution distributes the computational workload across multiple assistant nodes, ensuring higher fault tolerance and reliability. The authors implemented their proposal on real emulator devices,



(a) Two-party Access Control (b) Proxy-based Access Control

Figure 4.6: Designing Novel Proxy-based Access Control Scheme for Implantable Medical Devices [8].

In [9], the authors propose a lightweight attribute-based encryption (ABE) scheme to address security vulnerabilities in e-health edge computing environments. Their approach incorporates a modular exponential outsourcing technique, which reduces the computa-

tional load on constrained devices while enabling the outsourcing of shared data to an untrusted edge server. Figure 4.7, illustrates the detailed system model of their proposal. A powerful Edge Server (ECS) assists the Data Owner (DO) during the ciphertext generation phase. The DO needs to perform only a few multiplication calculations and some modular exponentiation calculations for encryption, significantly reducing their computational burden. To protect the privacy during outsourced computations, the authors introduce a masking algorithm and blind pairs to hide sensitive data. Despite these advantages, constrained devices still compute some complex calculations, which could pose challenges in highly resource-limited environments. Also, their approach does not consider the handling of urgent cases. This limitation could be critical in emergency situations where rapid access to vital health information is essential for timely and potentially life-saving interventions.

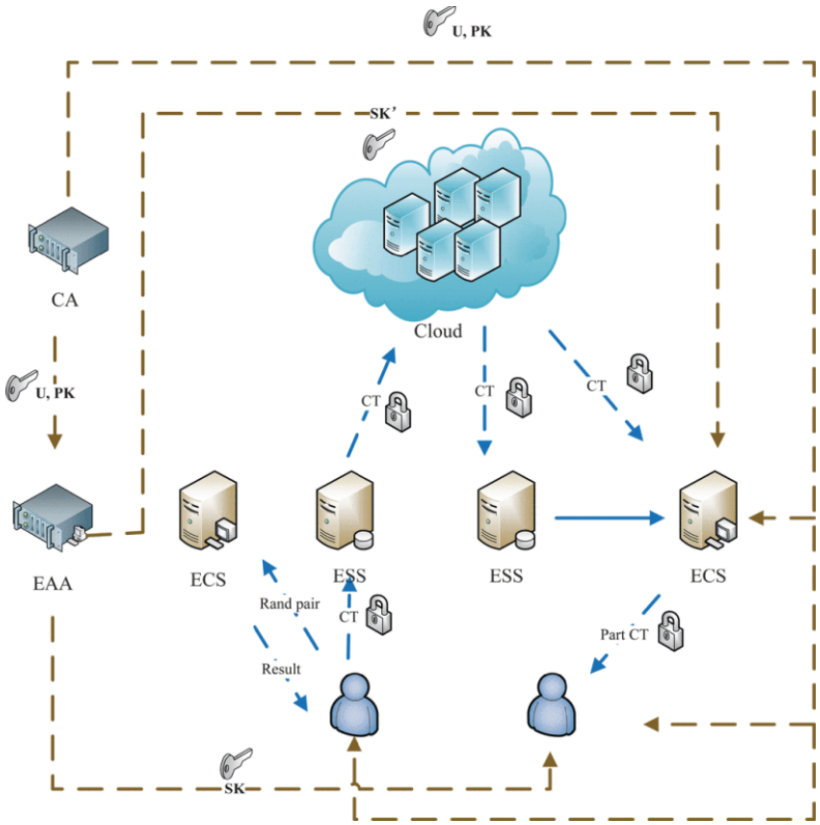


Figure 4.7: Secure Data Sharing With Lightweight Computation in E-Health [9].

In [10], the authors propose a fast CP-ABE system tailored for the healthcare domain. Their solution relies on outsourcing computationally expensive tasks to semi-trusted third parties, making it suitable for resource-limited devices. The proposed approach offloads the encryption, key generation, and decryption processes to a semi-trusted entity, as shown in Figure 4.8. The encryption process is divided into two phases, producing a ciphertext with two parts. In the first phase, the data owner (a constrained device) computes the initial part of the ciphertext, represented as:

$\tilde{C} = Me(g,g)^{\alpha s}, C = h^s,.$

The second phase is performed by the semi-trusted server, which completes the ciphertext generation. A significant limitation of this approach is that constrained devices are still mandated to perform exponentiation operations during ciphertext generation, which can be computationally demanding in resource-constrained environments.

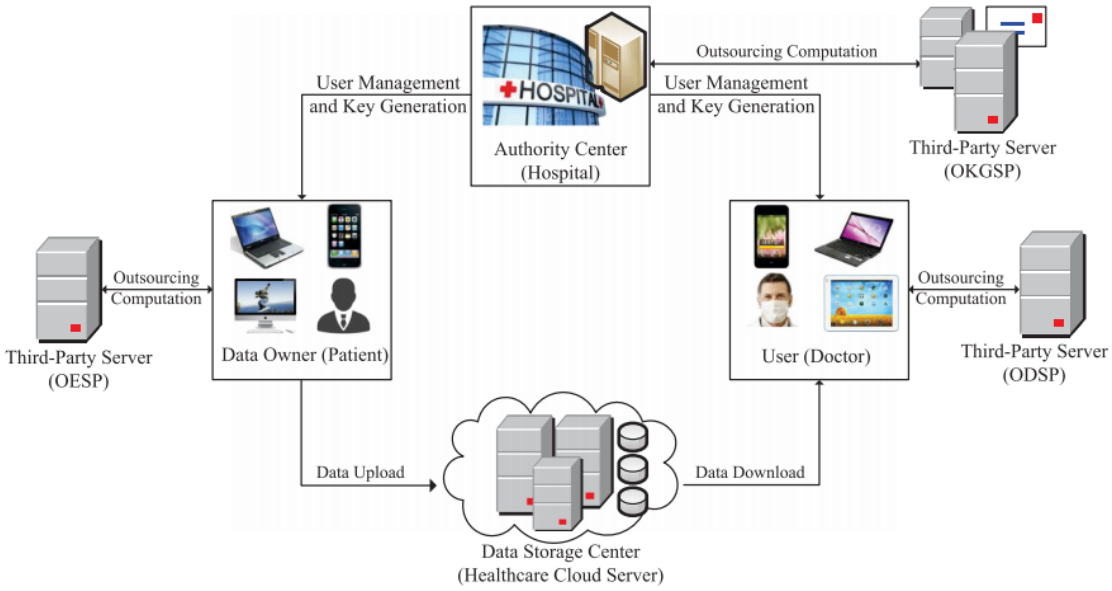


Figure 4.8: A fast CP-ABE system for cyber-physical security and privacy in mobile healthcare network System Model [10].

The authors in [88] propose modified fuzzy identity-based encryption (FIBE), a special case of ABE, schemes that require fewer optimal ate pairing operations compared to the original version. Additionally, they introduce modified Key-Policy Attribute-Based Encryption (KP-ABE) schemes that also reduce the number of pairing operations. To address the computational limitations of IoT devices, their approach outsources resource-intensive operations, such as scalar multiplication, exponentiation, and bilinear pairing, within the FIBE and KP-ABE schemes to high-powered servers. While effective, relying on high-powered servers for computation outsourcing can be relatively expensive. However, our solution offers the possibility of complementing these costly servers with more budget-friendly assistant nodes.

In [89], the authors propose a lightweight no-pairing ABE scheme based on Elliptic Curve Cryptography (ECC) to address the security and privacy issues in IoT. Their solution replaces the computationally expensive bilinear pairing operation with point scalar multiplication on an elliptical curve, making it lightweight and suitable for IoT devices. During encryption, the message M is protected using symmetric cryptography. The encryption key is generated from a random number using ECC, which can be reconstructed based on the attribute set. Their solution also ensures message integrity by using a hash

function. However, constrained devices are still required to perform some heavy computations to encrypt the message and verify its integrity, which remains a limitation in highly resource-constrained IoT environments.

4.3 Outsourcing, Load-Balancing, and Priority Based Solutions

In [11], the authors proposed a load-balancing solution for CP-ABE in an E-Health environment, based on a genetic algorithm to minimize the total encryption time. As shown in Figure 4.9, tasks are assigned to nearby constrained devices based on specific criteria such as resource availability, data size, access policy length, and task complexity. This approach aims to ensure an even distribution of workload across multiple nodes. However, a significant limitation of their method lies in the fact that the data is transmitted in clear text to the gateway, raising potential security concerns. The gateway executes the algorithm to distribute CP-ABE tasks among a group of devices. In contrast, our approach ensures enhanced security by transmitting data from sensors exclusively in encrypted format. This design guarantees that even the coordinator cannot access the data in its unencrypted form, thereby offering a more secure solution while maintaining load-balancing and efficiency.

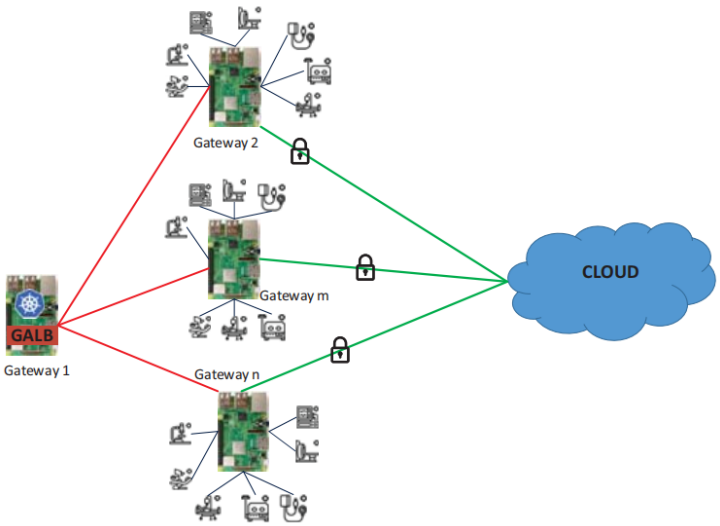


Figure 4.9: GALB: Load Balancing Algorithm for CP-ABE Encryption Tasks in E-Health Environment - Architecture [11].

In [90], the authors provided a comprehensive overview of various types of load balancers, including the network load balancer, the HTTP (S) load balancer, internal load balancer, hardware load balancer, software load balancer, virtual load balancer. They also examined key load-balancing measurement parameters such as throughput, average

response time, fault tolerance, scalability, performance, and resource utilization. Additionally, the study explored different load-balancing algorithms, including round Robin, least connections, weighted round Robin, source hash, least response time, and least bandwidth. The authors implemented round-robin and least-connect approaches in high-availability proxy servers, virtual machine clusters, and web servers. Their experiments involved testing these algorithms in real-world or simulated environments with high web traffic. The results revealed that among the algorithms evaluated, the round-robin algorithm outperformed the others in terms of overall performance.

We next review research papers that implement priority-based data handling. Our analysis emphasizes the methods used to determine data priorities, the benefits of adopting a priority-based approach, and the performance outcomes in terms of computational efficiency and energy consumption.

In a study by Bayrakdar et al. [91], a priority-based approach was introduced using IEEE 802.11af and sensor networks to monitor health data. The system categorized three priority levels: red (immediate need for advanced life support), yellow (observation for worsening condition), and green (non-urgent treatment). The proposed solution utilized sensor networks and opportunistic television white spectrum for health data transmission and was evaluated using Riverbed Modeler. Results confirmed the system's effectiveness in minimizing energy consumption and reducing delays.

In related work [92], propose a three-class priority packet scheduling algorithm tailored for large-scale Wireless Sensor Networks (WSNs) to handle real-time and non-real-time data efficiently. Their approach prioritizes tasks by classifying packets into three priority levels, giving emergency real-time tasks the highest priority to minimize end-to-end data transmission delays. Lower-priority packets are scheduled with fairness to avoid extended delays, even allowing the lowest-priority tasks to preempt the next level after a predefined wait time. Experimental results indicate that this approach outperforms traditional First-Come First-Served (FCFS) and multi-level queue schedulers in reducing transmission delays. This priority-based method demonstrates adaptability to dynamic WSN requirements, ensuring efficient and timely data handling.

In our work, we adopt a two-level priority scheme, categorizing messages into regular and urgent, rather than the three-level structure proposed in previous studies. While a three-level priority system can offer fine-grained task differentiation, it also introduces additional complexity in queue management and computational overhead, which may not be ideal for resource-constrained environments such as IoT. A two-level priority scheme offers a balance between performance and simplicity, enabling the system to efficiently prioritize critical real-time tasks (urgent messages) while still addressing non-urgent tasks (regular messages) without the overhead of managing an additional priority class.

4.4 Energy Consumption of ABE in IoT

The energy consumption of Attribute-Based Encryption (ABE) in IoT environments is a significant concern in IoT environments, due to the limited power and computational capabilities of IoT devices. This section examines existing studies that focus on optimizing ABE schemes to improve energy efficiency in IoT applications while ensuring security requirements.

In a study by Touati et al. [22], discussed in subsection 4.2, the proposed solution achieves significantly improved energy efficiency compared to traditional ABE methods by offloading all exponentiation operations from constrained devices. Recognizing exponentiation as the most energy-intensive operation in ABE encryption, this approach significantly reduces the device workload. Building on this approach, our solution further optimizes energy consumption by not only outsourcing heavy computations, but also implementing three sensor modes (active, idle, and sleep) and introducing a coordinator device to minimize communication exchanges. These enhancements are designed to further reduce energy usage and extend battery life.

In related work, Nzanywayingoma et al. [12] propose a secure CP-ABE method tailored for healthcare systems utilizing wireless body area sensors (WBASs). Their approach achieves constant-size ciphertext and constant computation costs, effectively addressing the challenges of security and energy efficiency in resource-constrained environments. By leveraging user credentials, the method ensures controlled access to sensitive personal health information (PHI), allowing only authorized entities to decrypt the data. While their solution reduces computational overhead, constrained devices are still required to perform some exponentiation operations. In contrast, our proposal completely eliminates exponentiation operations on constrained nodes, further optimizing energy consumption, and enhancing the scheme's suitability for IoT environments. Additionally, the authors highlight its suitability for healthcare applications by achieving lightweight encryption, reducing ciphertext size, and maintaining robust security, making it ideal for resource-limited devices.

In [93], the authors investigate the feasibility of adopting Attribute-Based Encryption (ABE) on constrained IoT devices, focusing on energy consumption and performance through real-world experiments. They implemented three ABE schemes from the literature [15] [57] [89] on ESP32 boards, a popular low-cost IoT platform, and evaluated their performance in terms of encryption/decryption time, energy usage, and battery life. The experiments, conducted with attributes ranging from 5 to 50, revealed that ABE operations impose significant computational and energy costs due to the limited memory and processing capabilities of constrained devices. However, these costs remain manageable when the number of attributes is kept low. The study also highlighted that increasing the number of attributes significantly reduces battery life, posing maintenance challenges such

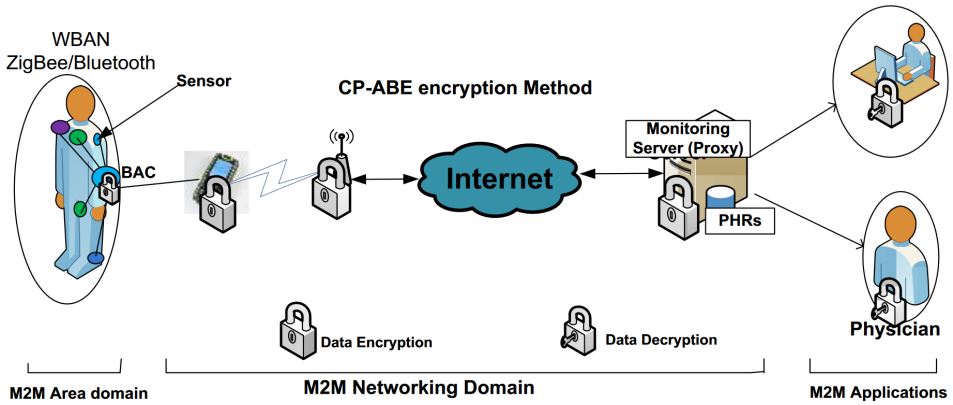


Figure 4.10: Improving Energy Efficiency in M2M Healthcare Systems Using CP-ABE Schemes [12].

as frequent battery replacements. Despite these challenges, the findings demonstrate that ABE is feasible for constrained IoT devices, especially with limited number of attributes. Building on these insights, our solution enhances ABE for constrained IoT devices by further optimizing energy efficiency and computational performance, making it more suited for resource-limited environments.

The extended work of [93], presented in [21], builds on the previous evaluation of ABE's feasibility for constrained IoT devices by incorporating additional platforms and innovative methodologies to improve the analysis. In this study, the authors implemented and tested three representative ABE schemes on two popular IoT platforms: ESP32 and RE-Mote. The inclusion of the RE-Mote platform, which features hardware support for cryptographic operations alongside very limited memory capabilities, provided a broader representation of IoT devices. The study evaluated the performance of ABE schemes under worst-case scenarios, considering up to 10 attributes in ciphertexts. It demonstrated that leveraging hardware cryptographic acceleration can significantly improve ABE's viability on resource-constrained devices, enabling acceptable battery lifetimes. A key contribution of this extended work is the introduction of a novel benchmarking method to estimate the average decryption performance in terms of time and energy consumption. This approach addresses the limitations of earlier evaluations that focused exclusively on worst-case scenarios, which tend to overestimate processing time and energy costs. The results reveal that average-case performance provides a more realistic perspective on the impact of ABE's on resource-constrained IoT devices. Overall, the energy consumption of ABE in constrained devices, such as ESP32 and RE-Mote, is significantly influenced by factors such as the complexity of encryption and decryption operations, the number of attributes involved, and the availability of hardware acceleration.

In [6], presented in Section 4.2 above, the authors proposed a dynamic smart offloading scheme that switches between full and partial encryption based on a decision tree model. They consider measuring the power consumption of partial and full encryption schemes

using a USB power meter, varying the number of attributes from 2 to 500 and testing two file sizes: 1 byte and 500 MB. The results demonstrated that energy consumption increases with the number of attributes in the access policy and is higher for larger files compared to smaller ones. However, the full encryption algorithm consumes more energy than the partial encryption algorithm in both scenarios. These findings highlight the benefits of computation outsourcing in IoT environments, as it significantly reduces energy consumption on resource-constrained devices, thereby enhancing their efficiency and extending operational longevity.

4.5 Comparative Study of Existing Solutions

In this section, we present a comparison between existing solutions and our proposed method. The Table 4.1 below summarizes the key differences and highlights the advantages of our approach. This comparison focuses on the most recent and relevant works, enabling a detailed analysis of their strengths and weaknesses. By identifying these aspects, we refine our approach and integrate the most effective features of each method to enhance our solution.

- **Year of Publication**: Specifies the year when the paper was published.
- **Suitability to IoT**: Assesses whether the solution is designed to meet the specific requirements of IoT environments.
- **Outsourcing Solution**: Indicates whether the solution involve outsourcing encryption tasks to reduce computational load on IoT devices.
- **Lightweight**: Refers to solutions that are efficient and optimized for low-power IoT devices with minimal resource usage.
- **Load-balancing**: Evaluates whether the solution incorporates load-balancing mechanisms to distribute workloads evenly.
- **Priority-based**: Identifies whether the solution includes mechanisms to prioritize urgent data for timely processing and management.
- **Availability of source code**: Highlights if the solution's source code is accessible online, essential for transparency and adaptability for future enhancements.
- **Healthcare Domain**: Indicates whether the solution is specifically tailored for applications in the healthcare sector.
- **Energy Consumption**: Assesses whether the solution takes energy efficiency into account in its design and implementation.

Ref	Year of Pub.	Suitable to IoT	Outsourcing Solution	Lightweight	Load Balancing	Priority Based	Source Code availability	Healthcare Domain	Energy Consumption
Bethencourt et al. [57]	2007	×	×	×	×	×	✓	×	×
Touati et al. [22]	2014	✓	✓	✓	×	×	×	×	×
Nzanywayingoma et al. [12]	2015	✓	×	✓	×	×	×	✓	✓
Yao et al. [89]	2015	✓	×	✓	×	×	×	×	×
Lounis et al. [3]	2016	✓	✓	✓	×	×	×	✓	✓
Zhao et al. [4]	2017	✓	✓	×	×	×	×	×	×
Nguyen et al. [7]	2018	✓	✓	✓	×	×	×	✓	✓
Girgenti et al. [93]	2019	✓	×	×	×	×	✓	✓	✓
Taha et al. [11]	2020	✓	✓	✓	×	×	×	✓	×
Taha et al. [6]	2020	✓	✓	✓	✓	×	×	×	✓
Zhang et al. [9]	2020	✓	✓	✓	×	×	×	✓	×
Wang et al. [10]	2020	✓	✓	×	×	×	×	✓	×
Zhong et al. [5]	2021	✓	✓	×	×	×	×	✓	×
Mahdavi et al. [88]	2023	✓	✓	✓	×	×	×	×	×
Wu et al. [8]	2024	✓	✓	✓	×	×	×	✓	×
Our Approach	2024	✓	✓	✓	✓	✓	✓	✓	✓

Table 4.1: Result of Comparative Study.

From the comparative analysis presented in the table above, it is observed that 94% of the solutions are *suitable for IoT* applications, while the remaining 6% may require further optimization to meet IoT-specific requirements. Regarding the *Outsourcing Solution* metric, 75% of the solutions leverage encryption outsourcing to offload heavy computations from IoT devices, improving their efficiency. For the *Lightweight* metric, 69% of the solutions are identified as lightweight, making them suitable for low-power IoT devices due to efficient resource usage, while the remaining 31% are more resource-intensive and better suited for higher-power devices. Concerning *Load-balancing* metric, 13% of the solutions incorporate load-balancing techniques to distribute tasks efficiently, whereas the remaining 87% either do not prioritize this factor or rely on a single entity (e.g., proxy, gateway, server, or edge node) for outsourcing. The *Priority-based* metric reveals that 6% of the solutions integrate priority-based management, while 94% do not address this aspect. For the *Availability of Source Code*, only 19% of the solutions provide access to their source code, a key feature for transparency and adaptability, while the rest do not. Under the *Healthcare Domain* metric, 62% of the solutions are explicitly tailored for healthcare applications. Lastly, in terms of the *Energy Consumption* metric, 37% of the

solutions evaluate and report energy consumption, while 63% neglect this critical factor, which is essential for resource-constrained IoT environments.

4.6 Conclusion

In this chapter, we provide a comprehensive review of existing solutions in the literature for lightweight Attribute-Based Encryption (ABE), focusing on various approaches and techniques developed to enhance the efficiency and security of ABE schemes. We analyze the strengths and limitations of each solution in comparison to our proposed approach. Considering the resource constraints in IoT environments, we identify the outsourcing of heavy computational operations as an effective strategy for achieving lightweight ABE. Furthermore, we evaluate these solutions in terms of computational efficiency and energy consumption, highlighting the advantages of our proposed solution over existing alternatives. In the following chapters, we will present our lightweight ABE scheme in detail, with a focus on load-balancing techniques to enhance outsourcing efficiency. Additionally, we will introduce our proposed energy model, specifically designed to minimize energy consumption and enhance the overall efficiency of IoT systems.

Chapter 5

Load-Balanced Attribute-Based Outsourced Encryption for Constrained IoT Devices

5.1 Introduction

In this chapter, we present our research on load-balanced attribute-based outsourced encryption for constrained devices and its impact on performance and energy efficiency [2]. This research focuses on healthcare applications, a promising area where IoTs are expected to make significant advancements.

In this work, we focus on the security aspect, particularly the confidentiality and access control of the data (for instance, EMR) during transmission and later after the data is stored in the cloud.

This chapter begins by demonstrating the issue using a healthcare example in the first section. In the second section, we will introduce our proposed system model for addressing these issues and define its components. Following that, we will explain our efficient distribution of heavy attribute-based encryption computation to multiple assistant nodes with varying capacities. In addition, our solution takes into consideration the case of heterogeneous sensors with different priorities, where urgent data should be encrypted first. Subsequently, we will introduce our energy consumption solution and the proposed energy model. Lastly, we will delve into the security analysis.

5.2 Problem Illustration for Healthcare Application

To illustrate the problem, we use healthcare applications as an example. As shown in Figure 5.1, medical sensors can be installed in a patient's body to monitor vital signals and send the data in the form of Electronic Medical Records (EMR) to the cloud for

storage, processing, and decision-making. As explained above, before sending sensitive data to the cloud, it should be encrypted (in our example, using CP-ABE). After that, the end-users (doctors, nurses, etc.) use their attributes to decrypt this data.

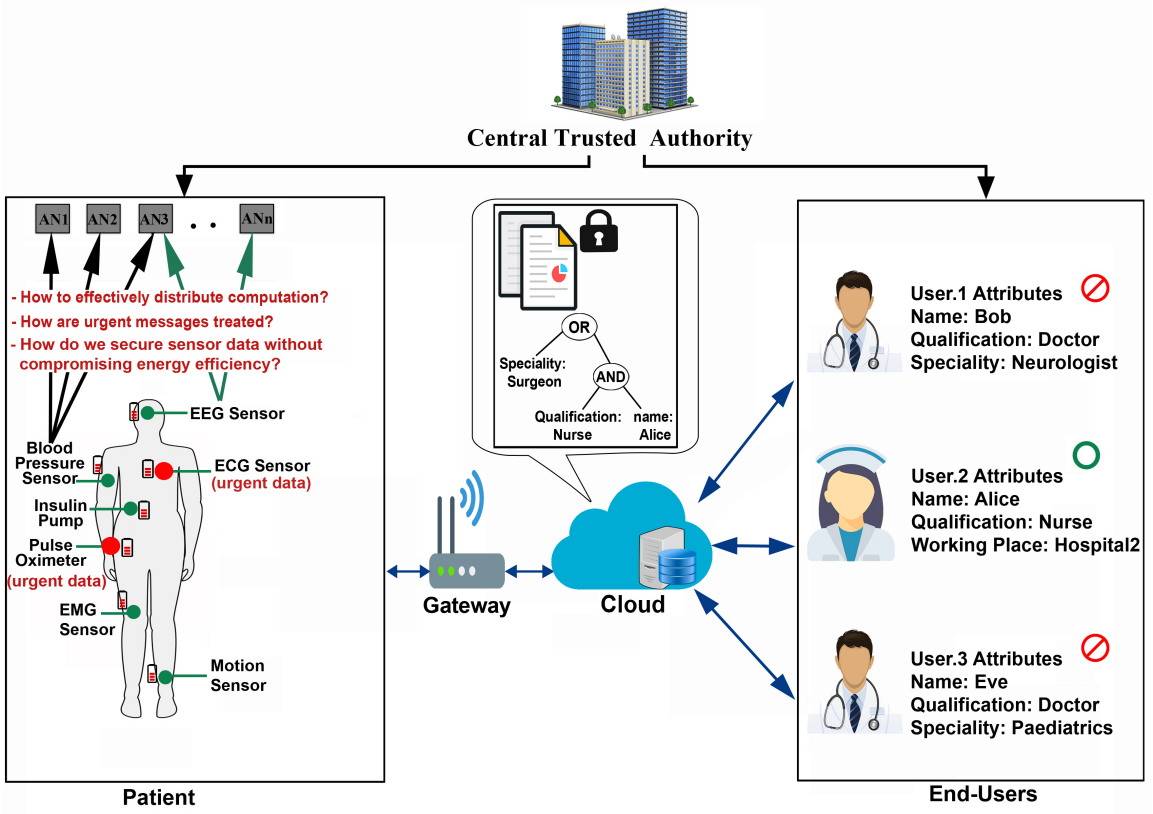


Figure 5.1: Problem Illustration for Healthcare Application

As we can see in Figure 5.1, seven sensors are installed in the patient's body, namely, EEG, ECG, Blood Pressure, Insulin Pump, Pulse Oximeter, EMG, and Motion.

In order to encrypt data using CP-ABE, each sensor has to compute a workload using Equation 2.3.

As we explained above, the number of exponents to be computed is very high. However, the sensors installed in the patient's body are very constrained devices and cannot support the execution of the operations of the CP-ABE.

Hence, several solutions have been proposed in the literature [94] with the aim of reducing workload and alleviating sensors. Some of them propose to outsource the heavy operations to a set of neighboring assistant nodes [11, 22] (see Figure 5.1).

Unfortunately, these solutions distribute the tasks to the assistant nodes equally without taking into consideration the nodes' capacities, which can decrease the performance of the system. Moreover, they did not take into account data priority, which is an essential factor in healthcare applications [91]. It shows how dangerously unwell the patient is based on the health data collected by sensors. In addition, the case where several sensors are installed in the patient's body is not also considered in the existing solutions. They

deal with only one sensor per patient.

To tackle the above challenges, we proposed in this work a framework that enables load-balanced attribute-based outsourced encryption for constrained IoT devices. The details are presented in the following sections.

5.3 System Model

In this section, we propose a solution for **RQ1**, and present our system model and the components needed to realize our solution in the context of healthcare applications. In general, a medical IoT system is composed of a set of resource-limited medical sensors (ECG, EKG, temperature, etc.) connected to the Internet via a gateway [95]. The sensors collected sensitive and private data, which must be encrypted. Only end-users who have access rights can decrypt the data.

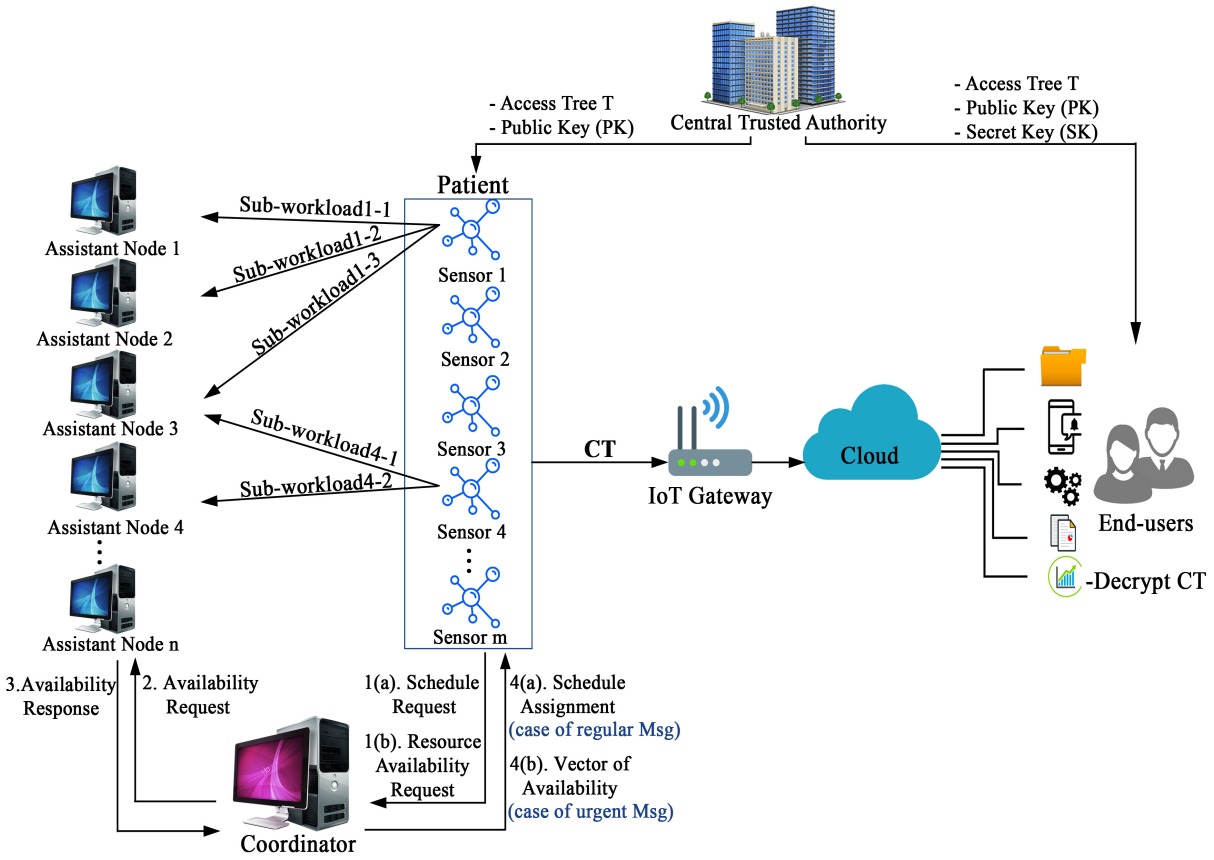


Figure 5.2: System Model.

The Figure 5.2 depicts our system model, which is composed of the following components: a Central Trusted Authority, a set of Sensors, a Coordinator, a set of Assistant Nodes, a Gateway, a cloud, and a number of End-Users. In the following sub-sections, we outline the functions and duties that each component performs.

5.3.1 Central Trusted Authority (CTA)

It is responsible for system initialization (setup, key generation, attributes, etc.). It generates and distributes public and master keys (PK, MK) to the sensors and end-users. In addition, the central trusted authority specifies and sends to both sensors and end-users the *access tree* T. The access tree is used for encrypting data by sensors and decrypting data by end-users when secret keys that are generated over a set of attributes satisfy this access tree.

5.3.2 Sensor Node (SN)

It represents the entity that gathers and sends its data to the cloud for storage. A patient is equipped with a set of wearable or implantable sensors. In order to encrypt a message M, the sensor uses the *public key* and the *access tree* T that is received from the Central Trusted Authority. As a result of the encryption, the sensor needs to generate the *ciphertext* CT (see Equation 5.1).

$$CT = (T, \tilde{C} = Me(g,g)^{\alpha S}, C = h^S, \\ \forall y \in Y : C_y = g^{q_y(0)}, C_y' = H(att(y))^{q_y(0)}) \quad (5.1)$$

As we explained above, though the sensor is a constrained device and cannot execute these heavy operations, it delegates the heavy computation to a set of trusted neighbor assistant nodes.

As illustrated in our system model (in Figure 5.2), m sensors want to encrypt messages with the help of n shared assistant nodes. However, this introduces a scheduling challenge, which led us to develop a solution incorporating a dedicated scheduler called the *Coordinator*. The primary function of the *Coordinator* is to effectively allocate assistant nodes to the sensors.

Besides, in our work, we assume that each sensor can encrypt two kinds of messages: urgent messages (with a high priority) and regular messages (with a low priority). The regular messages are predictable. The coordinator knows a priori the number of regular messages (given the number of messages and the number of sensor nodes), and it needs to check every time the status of assistant nodes and do the scheduling regularly. For example, we can do scheduling at 8 a.m. and 8 p.m. and send the list of available assistant nodes to the sensor that wants to encrypt its regular messages.

Thus, each sensor needs to encrypt a regular message, which it should first send to the coordinator as a scheduled request. As a response, it receives a schedule (as a vector) that represents the set of assigned resources. In our solution, a resource is considered as a CPU core of an assistant node. The sensor splits the workload according to the weight

values of the assigned resources (details are presented in the Algorithm 1).

As for urgent messages, the scheduling cannot be done by the coordinator, but it is rather a greedy scheduling done by the sensor node, which selects all the available assistant nodes. This sensor needs to first ask the coordinator for the list of available nodes.

When the sensor receives all the parts that are calculated by the assistant nodes, it uses them to calculate the *ciphertext* CT.

5.3.3 Coordinator

It is an additional device in the system. It performs the scheduling assignment to sensors, and it monitors the assistant nodes' states. After receiving the schedule requests from sensors, it retrieves the availability of assistant nodes and their weights. The coordinator employs distinct methods to assign schedules for regular messages and urgent messages. The role of the coordinator in the case of urgent messages is to reduce energy consumption by reducing the number of messages between the sensor node and the assistant nodes. For example, without a coordinator, the sensor node sends one message for each availability request and receives n messages (one from each assistant node). However, when we use the coordinator, the sensor node sends one message to the coordinator and receives one message from the coordinator. In the case of regular messages, the role of the coordinator is to ensure load balancing in addition to reducing energy consumption.

5.3.4 Assistant Node (AN)

In the system, we have a set of Assistant Nodes (AN) with different configurations. Their role is to perform heavy operations. Each assistant node receives the following from the sensor: PK, S_i, and $q_y^{(i)}(0)$. After that, it calculates:

$$\begin{aligned} C_{s_i} &= h^{S_i}, e(g,g)^{\alpha S_i} \\ \forall y \in Y : C_{y_i} &= g^{q_y^{(i)}(0)} \\ \forall y \in Y : C'_{y_i} &= H(att(y))^{q_y^{(i)}(0)} \end{aligned} \tag{5.2}$$

After finishing the calculation, they send the results to the sensor.

5.3.5 Gateway

Receives CT from the sensor and saves it in the cloud.

5.3.6 Cloud

The cloud storage serves as the central repository for storing encrypted data and access policies. It acts as a secure and scalable storage solution that allows authorized users

to access the data remotely (via a Web-based application). Cloud storage also provides high availability and redundancy for data, guaranteeing that it is always accessible and protected against data loss. Additionally, the cloud allows for efficient data sharing and collaboration among users, thus making it an essential part of the solution. For medical healthcare systems, cloud storage can be used to securely store and share sensitive patient information among healthcare professionals. Authorized end-users, such as doctors, nurses, and specialists, can access the data remotely while ensuring that it is encrypted and protected. This allows for efficient collaboration and decision-making while maintaining the confidentiality and integrity of patient data.

5.3.7 End-Users

Each user has the right to decrypt the CT that is sent by the gateway and should have a set of attributes γ satisfying the *access tree* T. In order to decrypt CT, each end-user performs the following:

1. Send the set of attributes γ to the Central Trusted Authority. The latter uses them to generate and send back the secret key SK.

2. Use PK, CT, and SK to decrypt and generate the message M.

5.4 Lightweight Distributed ABE Solution

In the previous section, we presented our system model, which describes the key components of our solution and their relationships. In this section, we present our lightweight distributed load-balancing encryption solution. It involves a series of activities carried out by the main participants in our system model, and they show the behavior of the system as a whole. Figure 5.3, depicts the overall solution as a flowchart.

First, we make the following assumptions:

- There is a set of assistant nodes with different capacities in the network.

- The assistant nodes are semi-trusted and do not conspire with each other.

- The coordinator and the gateway are entities within the network and are semi-trusted.

- The communications between the different entities in the system are encrypted using shared pairwise keys.

Below, we outline the key activities within our solution:

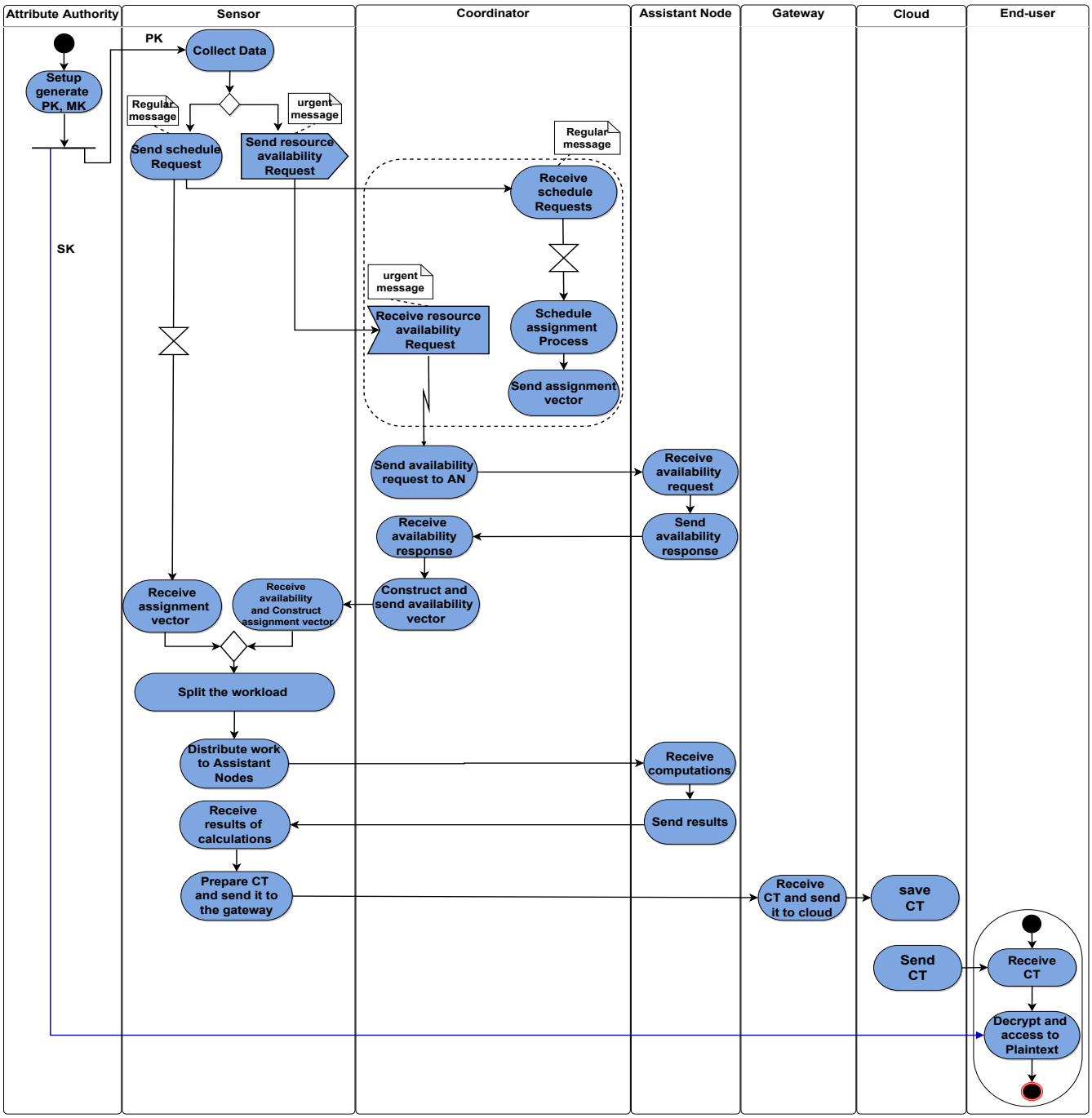


Figure 5.3: Proposed Lightweight Distributed ABE Solution.

5.4.1 Sensor Node: Requesting resources, distributing workload, and sending CT to the Gateway

In Algorithm 1, we illustrate the primary activities executed by each sensor node to communicate with the `Coordinator` and obtain a response providing the requested schedule.

Algorithm 1 Algorithm Run by Sensor Node: Requesting resources, distributing workload over Assistant Nodes, and sending CT to the Gateway

1: Input: Message M, n Assistant Nodes, $W = (w_i)_{(i=1,\dots,n)}$ where w_i is the processing power of Assistant Node i
2: Output : CT ▷ Output: Ciphertext
3: Let A be $A = (a_i)_{(i=1,n)}, a_i \in \mathbb{N}$ ▷ Assignment vector
4: Let R be $R = (r_i)_{(i=1,n)}, r_i \geq 0$ ▷ Available Resources at the ANs
5: Let Workload $\in \mathbb{N}$ ▷ Workload needed to encrypt M
6: Let Subworkload be Subworkload = $(\text{Subworkload}_j)_{(j=1,ar)}$ such that $\sum_{k=1}^{ar}$ Subworkload_k = Workload
7: **if** $(M.Priority = Low)$ **then** ▷ Regular Message
8: SN → Coordinator : ScheduleRequest()
9: Coordinator → SN : AssignmentVector(A) ▷ See Algorithm 2 below for details
10: **else** ▷ Urgent Message
11: SN → Coordinator : ResourcesAvailabilityRequest()
12: Coordinator → SN : ResourcesAvailabilityResponse(R) ▷ (see Algorithm 2 below for details)
13: ConstructAssignmentVector(A)
14: **end if**
▷ Split the Workload proportionally according to Vector A and W
15: Let Sum = $\sum_{k=1}^{n} w_k$
16: **for each** j in the range of j to ar **do**
17: $\text{Subworkload}_j \leftarrow$ Workload * w_j / Sum
18: **end for**
19: SN → AN : DistributeWork(Subworkload)
▷ Each Assistant Node performs computation assigned to it
20: AN → SN : Results()
21: PrepareCipherText(CT)
22: SN → Gateway : Send(CT)

There are two types of messages to encrypt by a sensor, regular and urgent messages. In the case of regular messages, the sensor sends a schedule request to the coordinator and waits to receive the assignment vector of resources (see Lines 7-9). In the case of urgent messages, the sensor sends an availability request to the coordinator and waits to receive all the available resources(see Lines 10-14).

In the case of regular messages, the sensor splits the workload according to the assignment vector and the weight values of assigned resources (see Line 14 to Line 18). The split takes as inputs i) the workload value, ii) the number of resources, and iii) the list of the corresponding weight values. We calculate the sum of the weights and use

each weight to determine the size of each sub-workload relative to the sum. In this way, the greater sub-workload should be assigned to the first resources (that have the highest weight values), and so on. We argue that this way improves the encryption time. Each sub-workload is sent to its corresponding assistant node, which calculates and returns the results to the sensor.

In the case of urgent messages, the sensor splits the workload among all the available resources. The sensor collects the results of the calculation from assistant nodes, prepares the final CT, and sends it to the Gateway, which saves it in the cloud. After transmitting the data to the cloud, end-users (such as doctors, nurses, etc.) can retrieve the finalized CT and use their attributes to decrypt this data. Only those who have a secret key with these attributes have the capability to decrypt the encrypted data and gain access to the plaintext.

5.4.2 Coordinator: Schedule Assignment

Algorithm 2, illustrates the sequence of activities carried out by the coordinator to assign schedules or resources to sensors. First, when the current time corresponds to one of the scheduled times, the coordinator waits for a certain time (T is defined during the system initialization) to receive all schedule requests (case of regular messages) from sensors (see Lines 6-8 Algorithm 2). After that, the coordinator sends availability requests to all assistant nodes and then prepares assignment vectors to be sent to the sensors. The assignment is performed based on the round-robin algorithm.

After that, it constructs a vector SR from these requests. The size of the vector corresponds to the number of sensors in the system. The negative values in the vector mean that the sensor has not sent a schedule request. Once all the schedule requests are received, the coordinator sends an availability request to all assistant nodes. It waits for a certain time to collect responses and construct a vector R from assistant node responses (see Lines 9 to 12). The size of this vector corresponds to the number of assistant nodes in the system. The negative values in the vector mean that all the resources of the assistant node are not available. When the vector has only negative values, this means that all the assistant nodes are not available. In this case, the coordinator places the received sensor requests in a queue, awaiting the release of resources from the assistant nodes. The available resources are divided and assigned equally to sensors that send requests (see Lines 13 to 23). For each received scheduling request, the coordinator constructs an assignment vector A. The size of this vector represents the number of assistant nodes. The positive values represent the number of assigned resources. Besides, from Line 25 to Line 33, we illustrate the activities that are performed by the coordinator in order to deal with urgent messages. The coordinator constructs the vector R of available resources in each available AN and sends it to the sensor.

Algorithm 2 Coordinator: Schedule Assignment

Input: n Assistant Nodes, m Sensor Nodes, $W = (w_i)_{(i=1,...,n)}$ where w_i is the processing power of Assistant Node i. T is the time to wait to receive schedule requests from sensor nodes
Output: $A = (A_k)_{(k=1,m)}, A_k = (a_i)_{(i=1,n)}, a_i \in \mathbb{N}$ ▷ for regular messages,
Output: Resources Availability ▷ for an urgent message
Let R be $R = (r_i)_{(i=1,n)}, r_i \geq 0$ ▷ Available Resources at the ANs
Let SN be SN $= \{sn_1, sn_2, ..., sn_m\}$ ▷ Sensor Nodes
Let SR be SR $= (sr_k)_{(k=1,m)}, sr_k \in \mathbb{N}$ ▷ Schedule Requests
if Current time corresponds to schedule request time **then**
▷ This is the case of regular messages
    WaitForScheduleRequests(T)
    SN → Coordinator: AllScheduleRequests(SR)
    ▷ After all SR are received from sensor nodes or T expires
    Coordinator → AN: AvailabilityRequest()
    WaitsForResponses()
    AN → Coordinator: AvailabilityResponse()
    ConstructVectorFromANAvailabilityResponses(R)
    ▷ R represents all available resources in the system
    Sort(R) ▷ Sort R according to W values in Descending Order
    ▷ Assign the available resources to sensors that send requests
    while $R > 0$ **do**
        for $sr_k \in$ SR **do**
            AllocateResourceToSN(R, sn_k)
            Update(R) ▷ Remove allocated resource from R
        end for
    end while
    for each sr_k in SR **do**
        ConstructAssignmentVector(A_k)
        Coordinator → sn_k : AssignmentVector(A_k)
    end for
end if
if Resource Availability Request is Received from Sensor Node k **then**
▷ This is the case of urgent message
    sn_k → Coordinator: ResourcesAvailabilityRequest()
    Coordinator → AN: AvailabilityRequest()
    WaitsForResponses()
    AN → Coordinator: AvailabilityResponse()
    ConstructVectorFromANAvailabilityResponses(R)
    Sort(R) ▷ Sort R according to W values in Descending Order
    Coordinator → sn_k: ResourcesAvailabilityResponse(R)
end if

5.5 Energy Savings Achieved Through Our Solution

In this section, we will demonstrate that our proposed solution effectively conserves sensor energy. We will examine the various operational modes of sensors and their impact on energy consumption, introduce our proposed energy model function, and conclude with a concrete example.

5.5.1 Modeling Operational Modes of Sensor Nodes

An effective approach to saving energy, optimizing usage and extending the operational lifespan of sensors in IoT networks involves implementing various modes, such as sleep/wake-up schemes [96, 97]. Sensor nodes can operate in different modes depending on the level of activity and requirements [98]. The sensor transitions between active, idle, and sleep modes to save energy during periods of inactivity. Figure 5.4 illustrates these modes and their transitions as a UML[1] State Machine Diagram, corresponding to the sensor states, which are instances of the class diagram shown in Figure 5.5. The transitions between these modes are determined by the specific tasks performed by the sensors.

1. **Active Mode**:
   - **Functionality**: In this mode, the sensor is fully operational. It processes, collects, and transmits data.
   - **Power Consumption**: The sensor consumes high power in this mode compared to other modes, as it is fully active.
   - **Use Case**: It is ideal for scenarios where real-time data is essential, such as the continuous monitoring of patients in critical condition within a hospital setting.

     In our solution, the main situations where the sensor stays in this mode are as follows:
     - Scheduled sensing and Data Collection
     - Transmitting requests/responses to/from coordinator
     - Preparing and distributing sub-workloads to assistant nodes while receiving their results.
     - Processing to generate ciphertext and transmit it to the cloud.
2. **Idle Mode**:
   - **Functionality**: In this mode, the sensor is partially powered and still able to detect some environmental changes or specific signals or inputs. The sensor

[1] Unified Modeling Language (UML): a standard modeling language

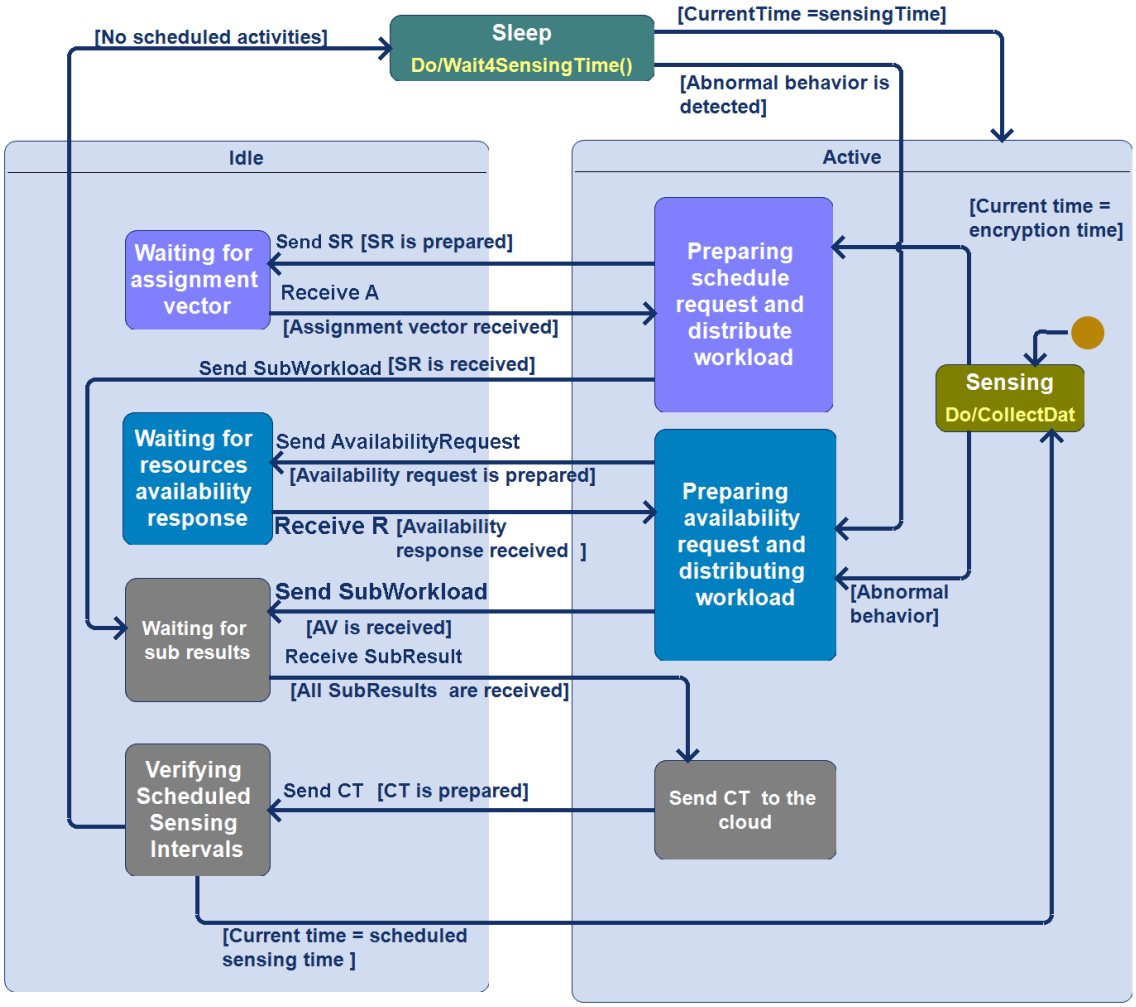


Figure 5.4: State Machine Diagram of Sensor Modes.

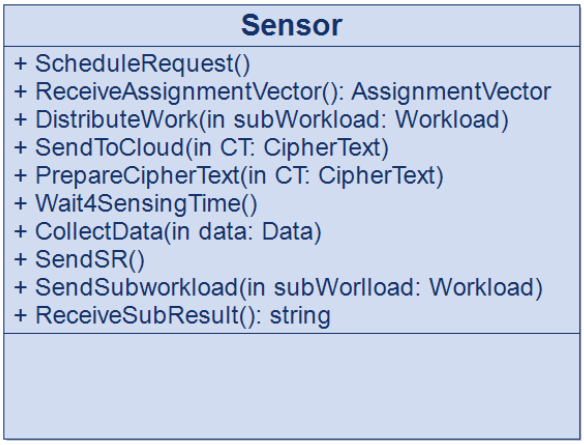


Figure 5.5: Sensor Class Diagram.

maintains only essential functions so that it can quickly return to full operational mode when needed.

- **Power Consumption**: The sensor consumes less power than the active mode,

as minimal components are powered.

- **Use Case**: It is used in low-power applications where systems require periodic checks or must wait for event triggers without fully activating. In our solution, the sensor remains in an idle state while waiting to receive responses from the coordinator or assistant nodes.

  For instance, a wearable glucose monitor for diabetic patients can remain in an idle state to conserve battery life and activate briefly at set intervals to measure glucose levels.

3. **Sleep Mode**:

   - **Functionality**: This mode allows the device to conserve power by shutting down most of its operations, with only essential components remaining active, such as circuits to detect specific wake-up triggers and the internal clock.
   - **Power Consumption**: Lowest power consumption, as compared to other modes of operation, as unnecessary components are turned off.
   - **Use Case**: Used for prolonged periods of inactivity or situations requiring high power savings, such as when the sensor only needs to wake up on a predetermined schedule or specific events (e.g., when motion is detected). For example, a wearable sensor on a patient recovering from surgery can remain inactive to conserve power, activating only if abnormal heart rate fluctuations are detected or if the patient has been stationary for an extended period or waiting for sensing time.

The transitions between the sensor modes are detailed below:

- **Active to Idle**:

  This transition occurs when the sensor has completed its tasks, such as data sensing and processing, but might need to resume these tasks again soon. In the context of healthcare application, the sensing is performed at regular intervals, the sensor can switch to idle mode after each data collection, conserving significantly more energy than remaining continuously in active mode. In addition, when the sensor needs to send requests or transmit data to the coordinator or the assistant nodes, it shifts to idle mode while waiting for a response.

- **Idle to Active**:

  It occurs when the sensor needs to perform tasks that require full power, typically in response to specific events or scheduled activities.

- **Idle to Sleep**:

  It occurs when the sensor does not anticipate any immediate tasks (e.g., the scheduled sensing time has not yet arrived and encrypted data have already been sent to the cloud), allowing it to conserve maximum energy.

- **Sleep to Active**: Typically triggered by specific events, scheduled activities (sensing in our case) or external commands.

5.5.2 Proposed Energy Consumption Model

In this section, we present the energy model of our system. Energy consumption is the total amount of power used by a system or device over a specific period of time. The formula for energy is expressed as follows:

$$E = P \times T \tag{5.3}$$

Where:

- **E** is the energy in joules (J)
- **P** is the power in watts (W)
- **T** is the time in seconds (s)

5.5.3 Objective Function of Energy

The objective function of energy consumption is defined as follows [99]:

$$\mathcal{F} = \sum_{t \in T} E^t_{Com} + \sum_{t \in T} E^t_{Cal} + \sum_{t \in T} E^t_{Sens} \tag{5.4}$$

The objective function $\mathcal{F}$ represents the total energy consumption of the sensor. This means that the total energy consumed by a sensor in an interval $[t, t+1]$ is composed of the following three components: the energy consumed for sensing (E^t_{Sens}), the energy consumed for communication (E^t_{Com}), and the energy consumed for calculation (E^t_{Cal}).

Where,

1. **Energy consumed in sensing**:

$$E^t_{Sens} = P_{sens} T_{sens} \tag{5.5}$$

Where:

- P_{sens} refers to the sensing power.

- T_{sens} refers to the sensing duration.

2. **Energy consumed in communication**:

$$E^t_{Com} = N \times (P_{tx}T_{tx} + P_{rx}T_{rx}) + P_wT_w \tag{5.6}$$

Where :

- P_{tx} represents the transmission power,
- P_{rx} represents the reception power,
- P_w represents the power in waiting state,
- T_{tx} represents the transmission duration,
- T_{rx} represents the reception duration,
- T_w represents the waiting duration.
- N represents the total number of exchanged messages during the encryption of a given data.

However, in our solution, the number of transmissions and receptions performed by the sensor are calculated as follows:

$$N = (M_c + \sum_{r \in R} M_r + M_{cloud}) \tag{5.7}$$

Where:

- M_c represents the number of messages exchanged with the coordinator, where in our solution:

  $M_c = 2$

- M_r denotes the messages exchanged with the assistant nodes within the range of assigned resources r, where R represents the assigned resources.
- M_c represents the number of messages exchanged with the cloud, where in our solution:

  $M_{cloud} = 2$

In our solution, we include the coordinator to minimize the messages exchanged during the check availability step between the sensors and assistant nodes. Reducing the number of messages will lower the frequency of data transmission and reception by the sensor, leading to energy conservation.

3. **Energy consumed in calculations**:

   The energy consumed during calculations is the energy used for computational tasks (multiplication, exponentiation, etc.). In our case, the sensor performs tasks such as splitting workloads and conducting multiple multiplications for preparing the ciphertext.

   The computation energy in $[t, t+1]$ can be expressed as follows:

$$E_{Cal}^{t} = K(P_{cal}T_{cal}) \tag{5.8}$$

   Where:

   - K represents the number of multiplication operations which is equal to the number of assigned resources (r)
   - P_{cal} refers to the power consumed for calculation,
   - T_{cal} representing the calculation duration,

5.6 Security Analysis

The security goal of our load-balanced attribute-based outsourced encryption is to provide fine-grained access control over encrypted data. In our solution, similar to standard ABE, access to encrypted data is determined by a set of attributes associated with the data and the attributes the end-users have. The primary aim is to ensure that only users who have the specified set of attributes can decrypt and access the encrypted data, thereby thwarting unauthorized access.

It is imperative that an attacker cannot retrieve the original message from the ciphertext without knowledge of the secret key. Hereafter, we illustrate how our approach withstands both passive and active attacks.

- **Resistance to passive attacks**: In a wireless channel, an attacker can intercept radio signals and eavesdrop on conversations between various nodes. This includes the exchange of information between coordinators and assistant nodes, coordinators and sensors, as well as communication among assistant nodes. Without encryption, sensitive data could be accessed without authorization as a result of this circumstance. In our work, we assume that all sensitive data is protected using symmetric encryption to defend against these passive wireless attacks. An adversary cannot decipher the ciphertext without access to the secret keys used in symmetric encryption.

- **Resistance to active attacks** :
  - **Masquerade Attack** The attackers may try to pretend to be legitimate users to gain unauthorized access to the data. However, with ABE, they cannot have access without having the appropriate attributes, which makes external attacks very difficult.
  - **Replay attack** : As the sensor communicates data regularly, all messages between the different entities of the system are timestamped, which prevents the attackers from capturing data and reusing it to gain unauthorized access.

5.7 Conclusion

In this chapter, we tackled the challenge of enabling IoT-constrained devices to efficiently use attribute-based encryption (ABE) by offloading intensive computational tasks to assistant nodes. This solution not only reduces the energy consumption but also significantly decreases computation time, making ABE more practical for resource-limited IoT devices. We applied our approach in the healthcare domain, where body sensors collect patient data and securely transmit it to the cloud via a gateway using attribute-based encryption. The solution achieved a reduction in energy consumption, improving the overall energy efficiency of the sensor network and extending sensor battery life, which enhances the system's reliability and lifespan. While our solution was demonstrated in healthcare, it is versatile enough to be applied in other areas such as environmental monitoring, military surveillance, and smart cities. In the next chapter, we will present the results of implementing our solution in terms of performance metrics and energy efficiency.

Chapter 6

Experimentation and Validation

6.1 Introduction

In this chapter, we will present the evaluation steps of our two contributions presented in the previous chapter. The first contribution focuses on the proposition of a new lightweight attribute-based encryption solution for IoT that minimizes execution time and ensures data priority, while the second contribution focuses on reducing the energy consumption of constrained devices through the use of different modes of operation techniques. We have developed our attribute-based encryption simulator, known as Load-Balancing ABE (LB-ABE-Sim) which we made freely available.

We have used this simulator to evaluate our proposal through a set of experiments. The obtained results show that our approach, compared to the existing outsourcing ABE solutions, improves computation time and energy consumption, ensures data confidentiality, and provides fine-grained access control.

6.2 Settings

To evaluate our solution, we have developed our attribute-based encryption simulator: Load Balancing ABE (LB-ABE-Sim), which we made freely available through github [1]. We conducted several numerical experiments in which we addressed the following questions:

- **RQ1**: What is the performance of our solution when we use a set of assistant nodes with different capacities?
- **RQ2**: To what extent the performance of our solution is affected when urgent messages are encrypted first?
- **RQ3**: How do we secure sensor data without compromising energy efficiency?

[1] https://github.com/loadbalancedabe/load.balanced.abe.simulator

In **RQ1**, we apply our approach and estimate the time spent during the encryption of several data gathered by a set of sensors installed in a patient's body. We study the effect of varying the number of assistant nodes and the effect of varying the number of sensors on the encryption time. As explained above in our approach, the workload is divided according to the assistant node's capacities. In this simulation, we have used configurations that are selected from real CPU models obtained from [2].

Besides, we run the same experiments but with three existing approaches. After that, we compared all the obtained results (in Section 6.4). These approaches are:

- **C-CP-ABE approach** [22]: the workload is divided equally.
- **Random approach**: We have the following two scenarios, and in each one, the workload is distributed randomly:
  - First, we sort the assistant nodes according to their capacities.
  - Second, we do not sort them.

As for the original implementation of C-CP-ABE's approach, they used only one sensor and a set of assistant nodes. To be able to compare our results, we updated the C-CP-ABE approach's implementation to take into consideration the case of several sensors in the patient's body in addition to the case of one sensor.

In **RQ2**, we conduct simulations, but in this case, we consider the potential inclusion of urgent messages (in **RQ1**, we used only regular messages, which are predictable, and the scheduling is performed by the Coordinator). In our approach, in the case of urgent messages, the scheduling is done by the sensor node, which selects all the available assistant nodes. But in the existing solutions, urgent messages are treated as regular messages. The two results are compared and presented in Section 6.5.

In **RQ3**, we measure the energy consumption of our proposed solution based on the obtained results of **RQ1**. We measure the energy of communication and compare it with the C-CP-ABE approach [22] to determine if our solution is more energy-efficient and also to highlight the importance of adding the coordinator into our system. The results are presented in Section 6.6. Next, we measure the energy of calculations and compare our obtained results with the CP-ABE approach [57] and [22]. Then we calculate the total energy consumption of our proposed solution by considering the communication, calculation, and sensing energy. This analysis allows us to demonstrate the overall efficiency of our system.

[2]Selected from real CPU models obtained from the links: https://www.cpubenchmark.net/cpu.php?cpu=Intel+Core+i7-6920HQ+%40+2.90GHz&id=2699 and https://setiathome.berkeley.edu/cpu_list.php

6.3 Experiment Metrics

6.3.1 Performance metric

In order to measure the performance of our solution, we measured the cumulative execution time that is spent during the encryption of messages. We take into account the CPU performance of the assistant nodes, which have different CPU models. Each CPU model has a set of resources (CPU cores). Each resource has a weight that represents floating point operations per second (flops).

It is important to note that our solution takes advantage of multiple cores, where the encryption program is multiple-threaded and can be parallelized and distributed across multiple cores. We assume that the CPU cores of a single machine are homogeneous, meaning they have similar performance capabilities.

Each workload is divided over a set of assigned resources and each of them finishes encrypting its sub-workload in a given time. To measure the encryption time for each sub-workload (Equation 5.2) on a given CPU model and a given core, we used the following formula:

$$EncryptionTime = \left(\frac{FPO}{FPC} \times CPO\right) + WT \tag{6.1}$$

where:

FPO: refers to Floating Point Operations. In our experimentation, it is estimated using the Equation 6.2.

$$FPO = ((S-1) \times 2) + \sum_{i=1}^{Y} ((q_{y0_i} - 1) \times 2) \tag{6.2}$$

FPC: refers to Flops Per Core (Core Weight)

CPO: refers to Cycles per Operation. It represents the average number of clock cycles required to complete a specific operation or instruction on a CPU. In this experiment, we suppose that CPO = 01.

WT: refers to the Waiting Time of the sensor so that its request will be processed by the coordinator. If all assistant nodes are not available, then, the coordinator places the received requests in a queue. The spent time in the queue should be added to the total encryption time.

Thus, the total encryption time for the workload corresponds to the slowest time among times spent for encrypting sub-workloads.

To show how to estimate the encryption time using Equation 6.1, we take an example of three sensors. We suppose that these sensors share three assistant nodes with the following capacities:

Assistant Node 1 (AN1) : Intel Core2 Quad Q6600 2.40GHz,

FPC = 2.44 GFLOPS, Number of cores = 4

Assistant Node 2 (AN2) : Intel Core i5-4210U 1.70GHz,

FPC = 2.46 GFLOPS, Number of cores = 4

Assistant Node 3 (AN3) : Intel Core i7-7700K CPU 4.20GHz,

FPC = 5.99 GFLOPS, Number of cores = 8

The coordinator sorts the assistant nodes based on their core weight (FPC) as follows: AN3, AN2, and then AN1.

When all the resources are available, the coordinator constructs the following three assignment vectors A1(3,1,2), A2(3,1,1), and A3 (2,2,1) respectively for the three sensors sn_1, sn_2, and sn_3.

However, when we use our approach, the sn_1 divides its Workload into six (number of assigned resources) parts which are not equal values. For instance, for the sake of illustration as an example, if the secret value S is equal to 100000000, then, the parts are as follows: 23666535, 23666535, 23666535, 9719478, 9640459, 9640458. In the same way, the sensor divides and distributes each q_{y0_i}.

Thus, the encryption time of the first sub-workload which is computed by a resource of AN_3 is estimated as follows:

$$EncryptionTime = (\frac{((23666535-1)\times 2)+\sum_{i=1}^{5}((q_{y0_i}-1)\times 2)}{5.99})\times 1) + 5 = 47412094\ ns$$

Approach	AN	S_i received by each Resource			ET per Resource (in ns)	Total ET (in ns)
		Resource 1	Resource 2	Resource 3		
Our approach	**AN3**	2321872	2321872	2321872	4651504	**4651504**
	AN2	953557	Free	Free	4651502	
	AN1	945804	945803	Free	4651498	
C-CP-ABE approach	**AN3**	1635130	1635130	1635130	3275724	**8041643**
	AN2	1635130	Free	Free	7976256	
	AN1	1635130	1635130	Free	8041643	
Random approach without sorting AN	**AN3**	590313	589857	142542	1843732	**24867212**
	AN2	1139717	Free	Free	6073595	
	AN1	4587883	2760468	Free	24867212	
Random approach with sorting AN	**AN3**	4196404	2971094	1484155	9753119	**9753119**
	AN2	527409	Free	Free	3407296	
	AN1	477272	154446	Free	1925104	

Table 6.1: Illustrative example of Workload distribution and estimated encryption times, where, $S = 9810780, |Y| = 5$

In Table 6.1, we present a comparison of the workload distribution values that are performed by sn_1 using the four approaches. In this workload, we have $S = 9810780$ and $|Y| = 5$.

Within this table, we demonstrate how the secret value (S) has been partitioned across the allocated resources, resulting in a set of S_i. As detailed earlier (as shown in Equation 5.2), each resource computes its respective sub-workload.

As we can see, the distribution is carried out by considering the nodes' capacities, while, in the C-CP-ABE approach, the sensor divides the Workload into equal values. The last column in the table represents the total estimated encryption time.

Comparing the total encryption time for the four approaches, we can observe that the encryption time is greatly improved when we use our approach. We get an improvement of around 73% compared to the C-CP-ABE approach, and more than 100% compared to the random approaches.

6.3.2 Energy Consumption Metric

We suppose a network of sensors and a set of assistant nodes range from 3 to 21. These sensors perform the operations of sensing, communication, and calculations. To compute

the energy consumed during the time interval $[t, t+1]$, we use the power values provided in Table 6.2 [99]:

Table 6.2: Used values of power for different modes.

Parameter	Value	Mode
Transmit (P_tx)	72 mW	Active
Receive (P_rx)	72 mW	Active
Sensing (P_sens)	3 μW	Active
Multiplication (P_mult)	0.5 μW	Active
Waiting (P_w)	10 μW	Idle
Sleeping (P_slp)	3 μW	Sleep

In the following, we use the values from Table 6.1 to demonstrate an example calculation of energy consumption using our model (presented in Sub-section 5.5.2).

1. **Energy Consumption Values for our approach:**

$$E^t_{Sens} = (0,000003 \times 120)$$
$$E^t_{Com} = N \times (0,072 \times 0.016 + 0,072 \times 0.016) + 0,000015 \times 0,004651504$$
$$N = (2 + (2 \times 3) + 2)$$
$$E^t_{Cal} = (3 \times (0,0000005 \times 0.01))$$

$F = 0.02340008477256\ (J)$
$F = 23.400084773\ (mJ)$

2. **Energy Consumption example with C-C-ABE approach:**

$$E^t_{Sens} = (0,000003 \times 120)$$
$$E^t_{Com} = N \times (0,072 \times 0.016 + 0,072 \times 0.016) + 0,00001 \times 0,008041643$$
$$N = (4 \times 3 + 2)$$
$$E^t_{Cal} = (3 \times (0,0000005 \times 0.01))$$

$F = 0.032616135624645\ (J)$
$F = 32.616135625\ (mJ)$

3. **Energy Consumption example with Random approach Without Sorting Assistant Nodes:**

$$E^t_{Sens} = (0,000003 \times 120)$$
$$E^t_{Com} = N \times (0,072 \times 0.016 + 0,072 \times 0.016) + 0,00001 \times 0,024867212$$
$$N = (2 + (2 \times 3) + 2)$$

$$E^t_{Cal} = (3 \times (0,0000005 \times 0.01))$$

$F = 0.02340038800818\ (J)$
$F = 23.400388008\ (mJ)$

4. **Energy Consumption example with Random approach With Sorting Assistant Nodes:**

$$E^t_{Sens} = (0,000003 \times 120)$$
$$E^t_{Com} = N \times (0,072 \times 0.016 + 0,072 \times 0.016) + 0,00001 \times 0,009753119$$
$$N = (2 + (2 \times 3) + 2)$$
$$E^t_{Cal} = (3 \times (0,0000005 \times 0.01))$$

$F = 0.023400161296785\ (J)$
$F = 23.400161297\ (mJ)$

6.4 Results for RQ1

1. **Effect of varying the number of sensors and assistant nodes:**

We present here the effect of varying the number of sensors and the number of assistant nodes on the performance of our solution in terms of encryption time. Since the results depend on the timings of different events, which are random, we repeated the same experiment several times and calculated the median for all the obtained encryption times. Besides, we vary the number of sensors, ranging from 1 to 15, and the number of assistant nodes, ranging from 3 to 21.

The results of encryption times are presented as a set of curves in Figure 6.1. These curves show the effect of varying the number of sensors and assistant nodes on the encryption time. The x-axis represents variation in the number of assistant nodes, while the y-axis is the median encryption time in nanoseconds.

We also varied the number of sensors from 1, 3, 6, 9, 12, and 15 sensors. The obtained results are depicted respectively in Figures 6.1a, 6.1b,6.1c,6.1d,6.1e,6.1f. As we can observe, in all cases, our approach gives the best encryption time (lowest values). In our approach, we split the workload based on assistant nodes' weights (capacity). Then, each assistant node received the appropriate workload. Thus, this allows a decrease in execution time.

The C-CP-ABE approach splits the workload equally among assistant nodes. The encryption takes more time than our approach. In the workload distribution, all assistant nodes take the same value, whereas the weak assistant nodes take a long

time to encrypt. On the other hand, we notice that the encryption time of the C-CP-ABE solution with or without sorting assistant nodes is almost the same because the workload is divided equally, and so all resources compute the same sub-workload.

In the Random approach, we employed a random distribution method to allocate workloads among assistant nodes. This leads to uneven workload distribution, whereby the heaviest sub-workload could be assigned to a less powerful assistant node while a more capable node receives a lighter sub-workload. This distribution imbalance has affected the overall encryption time. To ensure the reliability of the results obtained through the random approach, we repeated the workload distribution process multiple times and saved the encryption times for each iteration. We calculated the median of the obtained encryption times, which is less sensitive to outliers compared to other measures like the mean.

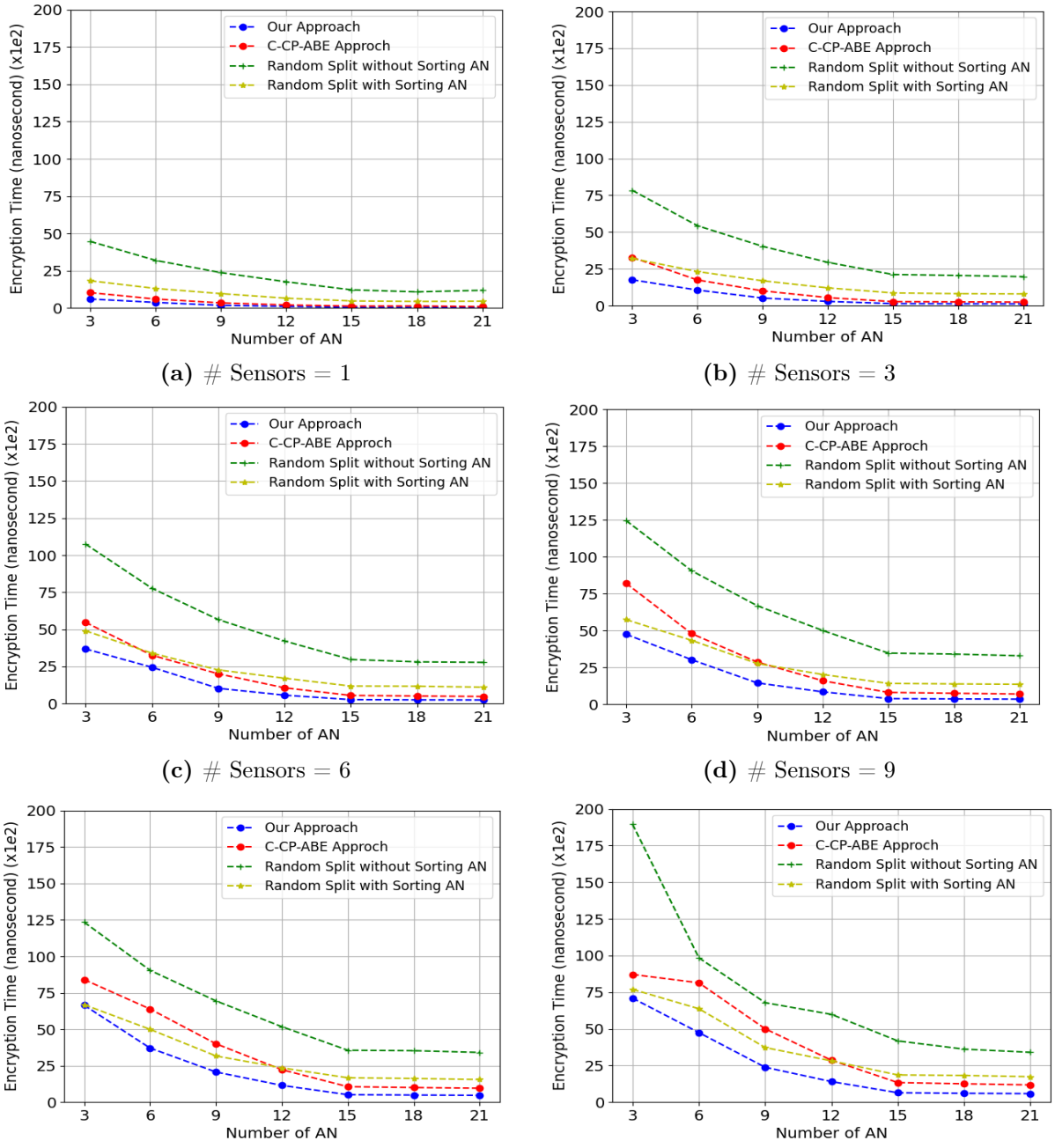
The curves clearly demonstrate that the worst results are obtained when we use a random approach without sorting nodes. This confirms our initial assumption that taking into consideration nodes' capacities in the workload distribution can significantly affect the performance of the solution.

2. **Effect of using homogeneous assistant nodes:**
In the previous experiment, we employed multiple assistant nodes with different capacities (heterogeneous). However, in this particular experiment, we focus on a set of assistant nodes with identical capacities (homogeneous assistant nodes) and compare the resulting encryption times. This enables us to verify and confirm that the capacity of the assistant nodes significantly influences the system's performance.

   To ensure robustness and accuracy, we conducted this experiment three times, changing the configuration of each assistant node. These three configurations include i) a set of weak assistant nodes, ii) a set of medium-capacity assistant nodes, and iii) a set of powerful assistant nodes. The obtained encryption times for each configuration are illustrated in Figure 6.2.

   As expected, our approach and the C-CP-ABE approach achieve similar encryption times values. This is explained by the fact that the workload is divided equally in our approach as the C-CP-ABE approach. The split in our approach is based on the assistant nodes' weights which are the same in this experiment. As for the random approach, the encryption time is always the worst compared to our values.

   All the results obtained from our experiments consistently validate our initial intuition that our solution significantly improves the performance of the system when compared to existing approaches (answer to RQ1).



(a) # Sensors = 1

(b) # Sensors = 3

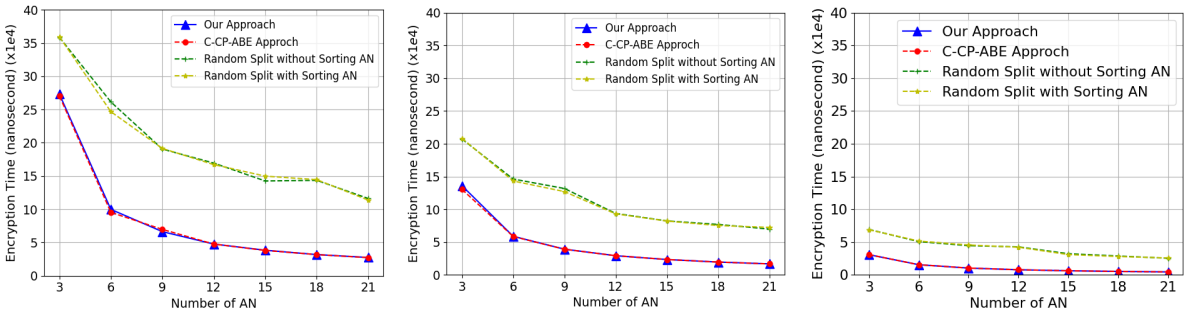
(c) # Sensors = 6

(d) # Sensors = 9

(e) # Sensors = 12

(f) # Sensors = 15

Figure 6.1: Effect of varying the number of assistant nodes and sensors.



(a) Weak ANs (FPC=2.0 GFLOPS).

(b) Medium ANs (FPC=3.26 GFLOPS).

(c) Powerful ANs (FPC= 6.18 GFLOPS).

Figure 6.2: Effect of using homogeneous assistant nodes (ANs), # Sensors = 12.

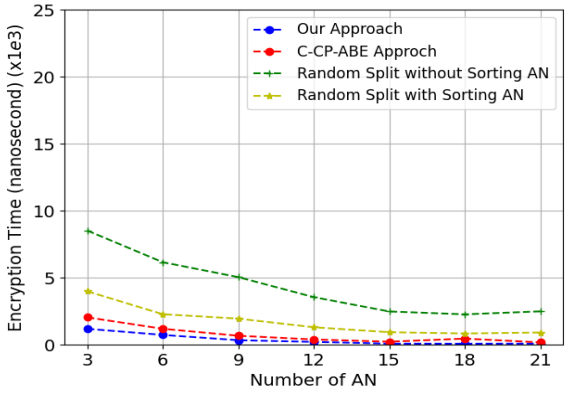
6.5 Results for RQ2

In this particular scenario, we introduce an additional consideration: the priority of the gathered data. This means that the sensor can transmit messages of different types, namely regular messages and urgent messages. However, we conducted the same experiment discussed in Section 6.4, but this time with a focus on urgent messages. Furthermore, we investigated the effects of varying the number of sensors and assistant nodes involved. Depending on the type of message received, the treatment of the data is adjusted accordingly. In our approach, we prioritize the processing of urgent messages over regular ones. In contrast, the other approaches treat urgent messages as if they are regular messages. The results of encryption time for the different approaches are presented in Figure 6.3.

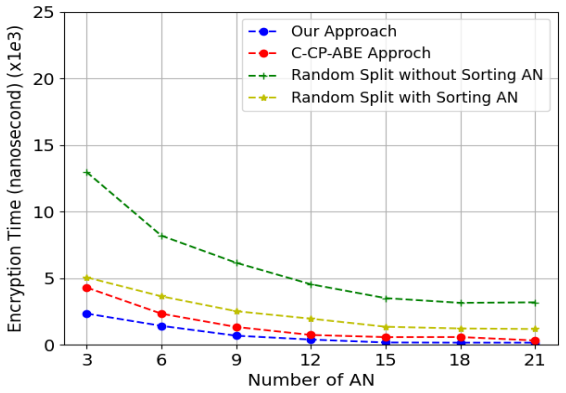
In our approach, even though we prioritize the encryption of urgent messages, this decision does not adversely affect the overall performance of the system. As we can observe, in all cases, our approach gives the best encryption times (lowest values) compared to other approaches (answer to RQ2). In addition, we notice that the encryption time increased compared to the results of Figure 6.1 (without considering urgent messages). In our method, we observe a growth rate of approximately 27% to 30%, which is better than the other approaches, namely, the C-CP-ABE approach shows an increase ranging from 29% to 91%. For Random split with sorting assistant node, the growth rate falls from 42% to 52%, while for Random split without sorting assistant node it ranges from 48% to 53%.

Besides, in healthcare applications, urgent messages should never be ignored, and sending them as fast as possible is crucial. Timely communication of urgent medical data can make a significant difference in patient outcomes and can even be a matter of life and death.

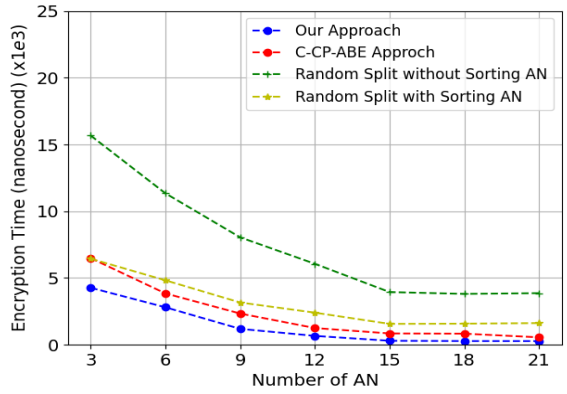
To assess the level of possible starvation with our proposal, we run additional simulations that provide more results, where we quantify the sensors' satisfaction rate, which reflects the rate of successfully served (encrypted and sent to the cloud) messages, both



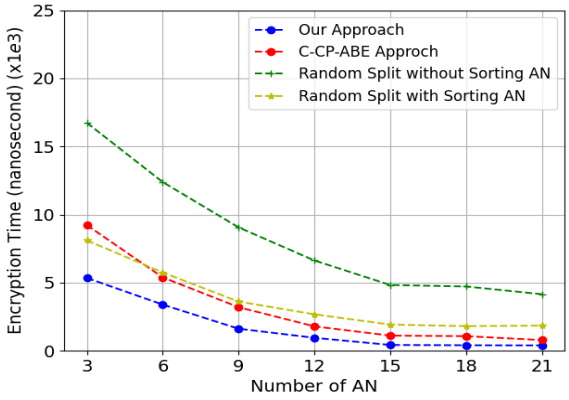
(a) # Sensors = 1



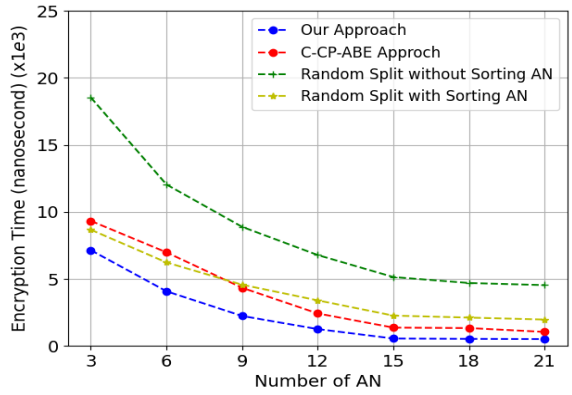
(b) # Sensors = 3



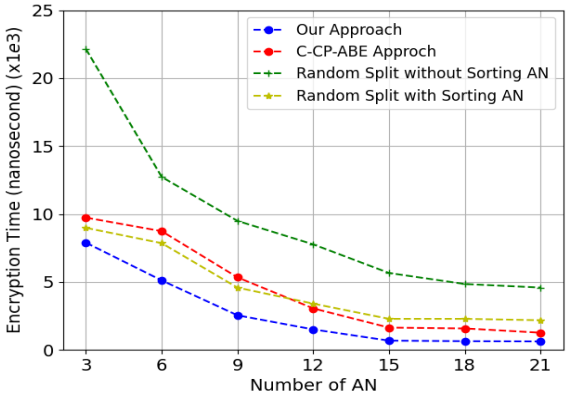
(c) # Sensors = 6



(d) # Sensors = 9



(e) # Sensors = 12



(f) # Sensors = 15

Figure 6.3: Effect of considering the urgent messages

urgent and regular.

We define $S_{satRate4U}$ as follows:

$$S_{satRate4U} = \frac{SU}{SU + NSU} \tag{6.3}$$

Where

$S_{satRate4U}$: refers to the satisfaction rate which is related to the rate of successfully served urgent messages.

SU: refers to the total number of successfully Served Urgent Sensor Requests

NSU: refers to the total number of Non Served Urgent Sensor Requests (out of patience time)

Furthermore, we quantify also the sensors' satisfaction rate ($S_{satRate4R}$) which measures the rate of successfully served regular messages. By adjusting the number of urgent messages, which allows us to determine whether our approach which is based on prioritize the urgent cases does not lead to starvation. This means that the regular messages should be also served and have a fair access to resources.

We define $S_{satRate4R}$ as follows:

$$S_{satRate4R} = \frac{SR}{SR + NSR} \tag{6.4}$$

Where

SR: refers to the total number of successfully Served Regular Sensor Requests.

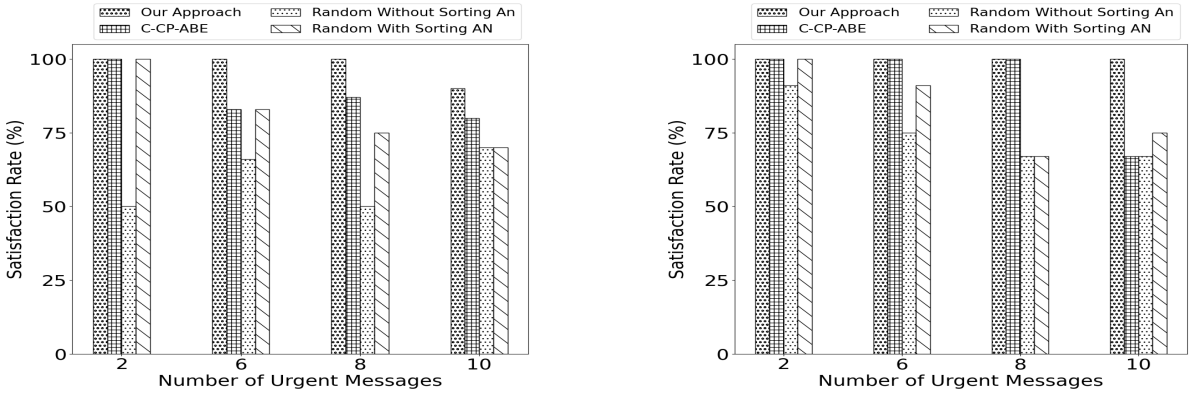
NSR: refers to the total number of Non-Served Regular Sensor Requests (out of patience time).

The obtained $S_{satRate4U}$ and $S_{satRate4R}$ values using the four approaches are depicted in Figure 6.4.

In this simulation, the number of urgent messages is randomized, and we explore the varying rates of these urgent messages. The total number of regular messages is fixed to 12.

As we can see, using our approach we always get 100% for the $S_{satRate4U}$ values, except in the last case where 9 urgent messages from 10 are encrypted successfully. This means that only one message is not encrypted because it exceeds the patience time in the queue of the coordinator. This is also related to the number of urgent messages, which is relatively high compared to the total number of messages (10/22 which represents 45%). Despite that, our approach gives the best $S_{satRate4U}$ scores compared to the other approaches, especially in the case of the random approach.

In scenarios where a random approach is employed without sorting assistant nodes (AN), we observe fluctuations in scores. For instance, the values of $S_{satRate4U}$ tend to rise as the number of urgent messages increases, ranging from two (02) to six (06) urgent messages. In this case, we observed that the secret value is divided, with heavier



(a) $S_{satRate4U}$ scores **(b)** $S_{satRate4R}$ scores

Figure 6.4: Sensors' Satisfaction Rates (#Sensors = 12, #AN = 6, #Regular message = 12).

sub-workloads being assigned to weaker assistant nodes, while lighter sub-workloads are allocated to more powerful assistant nodes. Consequently, the latter complete their calculations quickly and are then allocated to calculate other sub-workloads from other sensors. However, when new urgent messages arrive, they remain unattended as all assistant nodes have already been allocated. In the random approach, even with the inclusion of sorting assistant nodes, there is a slight improvement in results, but they remain unsatisfactory. This is related to the division of the secret value, which is performed without taking into consideration the capacities of assistant nodes.

As for the $S_{satRate4R}$ our approach gets 100% for the chosen scenario, which is not the case for random approach. This is due to our scheduling strategy, which is based on load balancing and ensures fair resource allocation. We believe also that in the case of medical and emergency applications, starvation even if it happens would not lead to extreme consequences given that regular messages are expected to be very low bandwidth and less sensitive to delays compared to urgent messages.

All these results demonstrate the effectiveness of our solution in terms of sensors' satisfaction rates, especially when the number of urgent messages is not greater than the number of regular messages (which is the normal case).

6.6 Results for RQ3

To evaluate the efficiency of our solution in terms of energy consumption, we need to measure the amount of energy used in communication, calculation, and sensing to accurately assess the overall impact. This involves accounting for the number of communications between the sensor and different entities along with the duration of each communication. We also consider the number and duration of the computational operations performed, enabling an accurate assessment of the overall energy impact.

1. *Case of Regular Messages:*

   We analyze the effect of varying the number of sensors and assistant nodes on the energy consumption of our solution. The results for the energy consumption during the interval $[t, t+1]$ are presented as a set of curves in Figure 6.5.

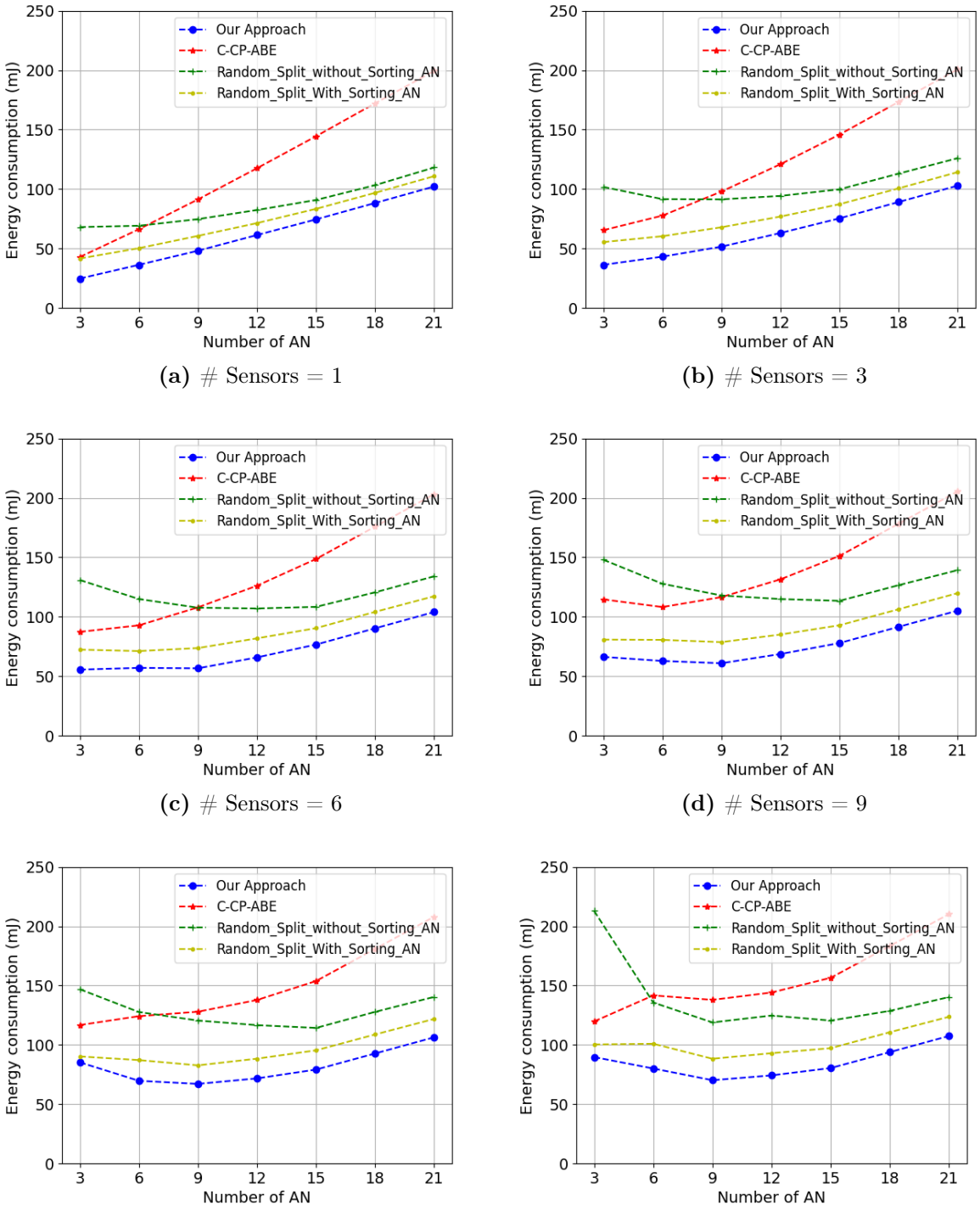
   The x-axis represents the number of assistant nodes, while the y-axis indicates the energy consumption in millijoule [mJ]. The number of sensors was varied across six values: 1, 3, 6, 9, 12, and 15 sensors. The corresponding results are depicted in Figures 6.5a, 6.5b, 6.5c, 6.5d,6.5e, 6.5f.

   As observed, our approach demonstrates superior energy efficiency, consistently achieving the lowest energy consumption across all scenarios. This efficiency is attributed to the shortest encryption time achieved by our approach, as demonstrated in the previous experiment. Consequently, the sensor experiences minimal waiting times compared to other approaches, which directly contributes to reducing overall energy consumption. In Figures 6.5d, 6.5e, and 6.5f, our approach shows a decrease in energy consumption from 3 to 9 assistant nodes, followed by an increase as the number of assistant nodes continues to grow. This demonstrates that the optimal number of assistant nodes for achieving the best energy consumption values lies within this range.

   We observed an increase in energy consumption as the number of assistant nodes grew, attributed to the rise in communication activities, which directly impacted energy usage. The C-CP-ABE approach exhibits the highest energy consumption among all methods, due to a greater number of communications and extended execution times that lead to longer sensor waiting periods and increased energy use.

   Although the Random approaches have the same number of communications as our approach, they consume less energy than the C-CP-ABE approach. However, the random workload distribution among assistant nodes in these approaches results in imbalances. Heavier sub-workloads may be assigned to less capable nodes, while more powerful nodes handle lighter sub-workloads. This imbalance increases the encryption time and consequently raises energy consumption. Notably, between 3 and 9 assistant nodes, the Random approach without sorting assistants results in higher energy consumption than the C-CP-ABE approach due to workload distribution without considering node capacities, leading to longer encryption times and increased energy consumption.

   The curves clearly illustrate that the highest energy consumption occurs with the C-CP-ABE approach. In contrast, our approach achieves superior energy efficiency, primarily due to the inclusion of a coordinator that minimizes the number of communications. This demonstrates the importance of considering node capacities in workload distribution. An optimized distribution reduces sensor waiting times, thereby



(a) # Sensors = 1 **(b)** # Sensors = 3

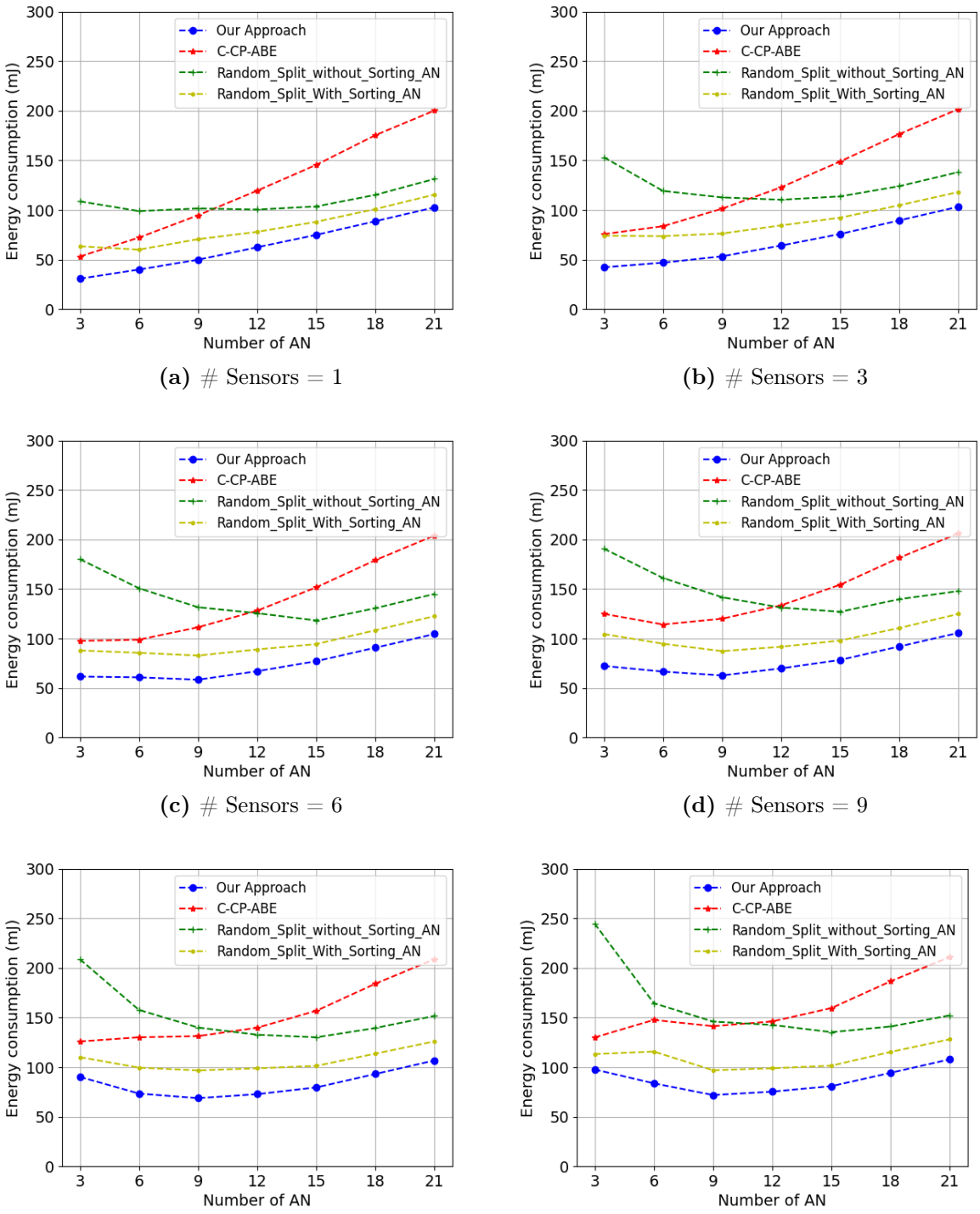
(c) # Sensors = 6 **(d)** # Sensors = 9

(e) # Sensors = 12 **(f)** # Sensors = 15

Figure 6.5: Energy Consumption of Sensors in Case of Regular Messages.

minimizing their overall active duration and energy consumption. Furthermore, the efficient transition of sensors between active, idle, and sleep modes in our approach plays a crucial role in reducing energy consumption. By allowing sensors to enter low-power modes during periods of inactivity, our solution significantly extends battery life while maintaining optimal performance, effectively balancing energy

efficiency and system reliability.



(a) # Sensors = 1

(b) # Sensors = 3

(c) # Sensors = 6

(d) # Sensors = 9

(e) # Sensors = 12

(f) # Sensors = 15

Figure 6.6: Energy Consumption of Sensors in Case of Considering Urgent Messages.

2. *Case of Considering Urgent Messages:*

In this experiment, we evaluated the performance of our approach under varying message priorities, including both regular and urgent messages. Sensors transmit-

ted data of different types, and we investigated the impact of prioritizing urgent messages on energy consumption, as well as the effects of varying the number of sensors and assistant nodes. The results, presented in Figure 6.6, demonstrate that our proposed approach consistently outperformed other existing methods in terms of energy efficiency. This improvement is attributed to the efficient coordination and optimized resource allocation enabled by our solution, which dynamically adjusts processing based on message priority. Unlike other approaches that treat all messages equally, our method prioritizes urgent messages without significantly compromising the processing of regular ones.

Despite the additional focus on urgent messages, our approach maintains the lowest energy consumption across all scenarios. While energy consumption increases compared to the results in Figure 6.5 (scenarios without urgent messages), the growth rate for our method remains between 0.40% and 24.02%, which is significantly lower than that of other approaches. In contrast, the C-CP-ABE approach exhibits increases ranging from 0.43% to 98,32%, the Random split with sorting assistant nodes shows a growth rate between 3,4% and 52,21%, and the Random split without sorting assistant node ranges from 6,18% to 59,75%. These results highlight the effectiveness of our method in balancing energy efficiency and responsiveness, making it particularly suitable for scenarios requiring real-time decision-making and efficient handling of prioritized tasks.

6.7 Threats to Validity

Outsourcing encryption operations to assistant nodes has been proposed in a number of contributions in the literature, which have been validated in many peer-reviewed papers [3, 6, 8, 11, 22]. Our main contribution in this paper is devising a new method that performs outsourcing in an improved way, taking into consideration the heterogeneity of assistant nodes and urgent messages. Therefore, the main focus is on showing the obtained performance with the new solution rather than on proving the validity of the general outsourcing scheme. Simulations also allow us to perform controlled experiments, which are based on manipulating (controlling) one or more variables while keeping other variables constant and measuring the effects on outcome variables. This control is often difficult to achieve in experiments involving real patients due to various factors, such as ethical considerations and patient safety. In addition, conducting experiments, at an early stage of solution validation, on the bodies of patients can be extremely expensive due to the need for medical personnel, equipment, and potential complications. Note that although simulations allow us to capture the global overview of the expected results and is the first step to validate our solution, real implementation and prototyping are needed to provide more accurate results, and this should be considered in future works.

Regarding the generalization of the applicability of our proposal, our model is designed to be independent of any specific context. Our solution can be reused in contexts beyond healthcare, provided that there is a coordinator, a set of assistant nodes, and a set of sensors, all running appropriate software. Therefore, there is no limitation beyond the previously mentioned requirements, which makes our solution applicable in many contexts.

6.8 Discussion & Achievements

The following aspects are covered in our scheme's security model:

1. **Lightweight**
   The outsourcing of heavy encryption operations to assistant nodes reduces the load for constrained devices. Thus, this may decrease the energy consumption of the restraining devices.

2. **Load-balancing of workload**
   Using the load-balancing technique, the outsourced operation is allocated equally between the assistance nodes. This may shorten the time required to complete an encryption process.

3. **Confidentiality**
   The data's content must be kept private. Users who have not been designated as receivers by the data owner should be denied access to the data.

4. **Fault Tolerance** A system can continue performing its expected function in the presence of faults. The use of a load-balancing algorithm ensures this feature. When a node fails, another node receives the request automatically. This is because the coordinator checks the status of the assistant nodes every time.

5. **Data priority**
   The priority of the data should be considered, particularly in the medical area. There are two levels of priority: regular and urgent.

6. **Energy consumption**
   Improved thanks to the coordinator that is used in our solution.

6.9 Conclusion

In this chapter, we evaluated our proposed solution through simulations across various scenarios and compared its performance with existing approaches. The results clearly demonstrate that our approach significantly enhances system performance and optimizes

resource allocation compared to previous solutions. Notably, it achieves a substantial reduction in the energy consumption of resource-constrained devices. Moreover, our solution achieved the best satisfaction rate scores and minimized encryption time for urgent data, a critical requirement in applications such as healthcare, where timely and efficient data processing is essential. The integration of a coordinator into our approach proved particularly impactful, delivering notable improvements in energy efficiency and execution time. These findings underscore the effectiveness of our solution in addressing the challenges of IoT environments. The next chapter will present the general conclusion of this thesis and outline potential directions for future research to build upon the findings of this work.

General Conclusion

This thesis addressed the growing need for secure and flexible access control mechanisms in Internet of Things (IoT) environments. The primary objective of this research was to design a lightweight version of Attribute-Based Encryption (ABE) tailored to the constraints of IoT devices, optimizing execution time and energy consumption without compromising security. The proposed lightweight ABE model demonstrated significant improvements in access control performance for IoT networks while minimizing the execution time and energy usage. The experimental results confirmed that the model not only outperformed existing ABE solutions in terms of computational efficiency but also maintained robust security levels. Moreover, it prioritized critical data, an essential feature for applications in domains like healthcare, where timely and secure data handling is paramount.

This research makes both theoretical and practical contributions to the advancement of ABE mechanisms. Theoretically, it presents an adapted ABE framework that ensures data security, access control, and prioritization in healthcare applications. It also explores an outsourcing technique to offload resource-intensive computations from healthcare devices, enhancing overall system performance. A load-balanced strategy was implemented to distribute tasks across different assistant nodes based on their capacities, leading to better execution time and more efficient use of resources. Practically, the effectiveness of the proposed framework was validated through simulations and experiments, demonstrating its potential applicability across various IoT domains. The framework provides a robust foundation for secure and efficient access control, making it viable for real-world implementation.

Despite these achievements, the research has certain limitations. First, the model was evaluated in a controlled, simulated IoT environment. Its performance in large-scale, real-world IoT ecosystems, which feature unpredictable traffic patterns and diverse security threats, remains to be assessed. Second, the study focused on Ciphertext-Policy ABE (CP-ABE), which is particularly suited for healthcare IoT applications. Future research should explore other ABE variants, such as Key-Policy ABE (KP-ABE), to determine their suitability for broader IoT use cases.

Future directions:

1. **Real-World Testing**: Evaluate the proposed lightweight ABE model in live healthcare IoT systems to assess its practicality, scalability, and resilience against real-

world challenges.

2. **Expanding to Other Domain**s: Extend the application of the model to other critical or non-critical domains, such as finance, government, and education, to gauge its adaptability.

3. **Incorporating KP-ABE**: Adapt and test the model with KP-ABE to analyze its impact on system performance, flexibility, and security.

4. **Policy and Attribute Updates**: Explore methods for dynamic access policy and attribute updates to enhance the model's adaptability in evolving IoT ecosystems.

In conclusion, this thesis provides a robust and scalable solution to the challenges of implementing secure and efficient access control in IoT environments. By addressing the limitations of existing ABE models and proposing a lightweight attribute-based encryption approach, this work contributes significantly to the development of secure IoT systems. It is anticipated that the findings and methodologies presented will inspire further research and innovation in the field, advancing the adoption of secure, scalable, and efficient access control mechanisms to meet the demands of rapidly evolving IoT technologies.

Bibliography

[1] Ovidiu Vermesan, Peter Friess, Patrick Guillemin, Sergio Gusmeroli, Harald Sundmaeker, Alessandro Bassi, Ignacio Soler Jubert, Margaretha Mazura, Mark Harrison, Markus Eisenhauer, et al. Internet of things strategic research roadmap. In *Internet of things-global technological and societal trends from smart environments and spaces to green ICT*, pages 9–52. River Publishers, 2022.

[2] Meriem Gasmi, Mohamed Lamine Kerdoudi, and Abdelmalik Bachir. Load-balanced attribute-based outsourced encryption for constrained iot devices. *Computers and Electrical Engineering*, 118:109424, 2024.

[3] Ahmed Lounis, Abdelkrim Hadjidj, Abdelmadjid Bouabdallah, and Yacine Challal. Healing on the cloud: Secure cloud architecture for medical wireless sensor networks. *FGCS*, 55:266–277, 2016.

[4] Zhiyuan Zhao and Jianhua Wang. Verifiable outsourced ciphertext-policy attribute-based encryption for mobile cloud computing. *TIIS*, 11(6):3254–3272, 2017.

[5] Hong Zhong, Yiyuan Zhou, Qingyang Zhang, Yan Xu, and Jie Cui. An efficient and outsourcing-supported attribute-based access control scheme for edge-enabled smart healthcare. *Future Generation Computer Systems*, 115:486–496, 2021.

[6] Mohammad Bany Taha, Hakima Ould-Slimane, and Chamseddine Talhi. Smart offloading technique for cp-abe encryption schemes in constrained devices. *SN Applied Sciences*, 2(2):274, 2020.

[7] Kim Thuat Nguyen, Nouha Oualha, and Maryline Laurent. Securely outsourcing the ciphertext-policy attribute-based encryption. *World Wide Web*, 21(1):169–183, 2018.

[8] Longfei Wu and John Du. Designing novel proxy-based access control scheme for implantable medical devices. *Computer Standards & Interfaces*, 87:103754, 2024.

[9] Leyou Zhang, Xuehuang Gao, and Yi Mu. Secure data sharing with lightweight computation in e-health. *IEEE Access*, 8:209630–209643, 2020.

[10] Shulan Wang, Haiyan Wang, Jianqiang Li, Huihui Wang, Junaid Chaudhry, Mamoun Alazab, and Houbing Song. A fast cp-abe system for cyber-physical security and privacy in mobile healthcare network. *IEEE Transactions on Industry Applications*, 56(4):4467–4477, 2020.

[11] Mohammad Bany Taha and Rasel Chowdhury. Galb: Load balancing algorithm for cp-abe encryption tasks in e-health environment. In *Proceeding of IEEE ICRCICN*. IEEE, 2020.

[12] Frederic Nzanywayingoma and Huang Qiming. Improving energy efficiency in m2m healthcare systems using cp-abe schemes. In *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, pages 1243–1248. IEEE, 2015.

[13] Michael Chui, Mark Collins, and Mark Patel. The internet of things: Catching up to an accelerating opportunity. 2021.

[14] Fei Hu. *Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations*. CRC Press, 2016.

[15] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98, 2006.

[16] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *Annual international cryptology conference*, pages 213–229. Springer, 2001.

[17] Seyyed Keyvan Mousavi, Ali Ghaffari, Sina Besharat, and Hamed Afshari. Security of internet of things based on cryptographic algorithms: a survey. *Wireless Networks*, 27(2):1515–1555, 2021.

[18] P Sarwesh, N Shekar V Shet, and K Chandrasekaran. Energy-efficient network architecture for iot applications. In *Beyond the Internet of Things: Everything Interconnected*, pages 119–144. Springer, 2017.

[19] Musa G Samaila, Miguel Neto, Diogo AB Fernandes, Mário M Freire, and Pedro RM Inácio. Security challenges of the internet of things. *Beyond the internet of things: Everything interconnected*, pages 53–82, 2017.

[20] Senthil Kumar Jagatheesaperumal, Preeti Mishra, Nour Moustafa, and Rahul Chauhan. A holistic survey on the use of emerging technologies to provision secure healthcare solutions. *Computers and Electrical Engineering*, 99:107691, 2022.

[21] Pericle Perazzo, Francesca Righetti, Michele La Manna, and Carlo Vallati. Performance evaluation of attribute-based encryption on constrained iot devices. *Computer Communications*, 170:151–163, 2021.

[22] Lyes Touati, Yacine Challal, and Abdelmadjid Bouabdallah. C-cp-abe: Cooperative ciphertext policy attribute-based encryption for the internet of things. In *Proceeding of INDS conference*. IEEE, 2014.

[23] Marco Rasori, Michele La Manna, Pericle Perazzo, and Gianluca Dini. A survey on attribute-based encryption schemes suitable for the internet of things. *IEEE Internet of Things Journal*, 2022.

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

[25] Gurpreet Singh Matharu, Priyanka Upadhyay, and Lalita Chaudhary. The internet of things: Challenges & security issues. In *2014 International Conference on Emerging Technologies (ICET)*, pages 54–59. IEEE, 2014.

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

[27] Wan-Soo Kim, Won-Suk Lee, and Yong-Joo Kim. A review of the applications of the internet of things (iot) for agricultural automation. *Journal of Biosystems Engineering*, 45:385–400, 2020.

[28] Jinyuan Xu, Baoxing Gu, and Guangzhao Tian. Review of agricultural iot technology. *Artificial Intelligence in Agriculture*, 6:10–22, 2022.

[29] Rayes Ammar and Salam Samer. *Internet of Things From Hype to Reality: The Road to Digitization*. Spinger, 2019.

[30] Michael Chui, Markus Loffler, and Roger Roberts. The internet of things. Technical report, McKinsey Global Institute, 2010.

[31] Yuanhao Cui, Fan Liu, Xiaojun Jing, and Junsheng Mu. Integrating sensing and communications for ubiquitous iot: Applications, trends, and challenges. *IEEE Network*, 35(5):158–167, 2021.

[32] Abhishek Khanna and Sanmeet Kaur. Internet of things (iot), applications and challenges: a comprehensive review. *Wireless Personal Communications*, 114:1687–1762, 2020.

[33] S Narasimha Swamy and Solomon Raju Kota. An empirical study on system level aspects of internet of things (iot). *IEEE Access*, 8:188082–188134, 2020.

[34] Damilola Oladimeji, Khushi Gupta, Nuri Alperen Kose, Kubra Gundogan, Linqiang Ge, and Fan Liang. Smart transportation: an overview of technologies and applications. *Sensors*, 23(8):3880, 2023.

[35] Yazdan Ahmad Qadri, Ali Nauman, Yousaf Bin Zikria, Athanasios V Vasilakos, and Sung Won Kim. The future of healthcare internet of things: a survey of emerging technologies. *IEEE Communications Surveys & Tutorials*, 22(2):1121–1167, 2020.

[36] Mostafa Haghi Kashani, Mona Madanipour, Mohammad Nikravan, Parvaneh Asghari, and Ebrahim Mahdipour. A systematic review of iot in healthcare: Applications, techniques, and trends. *Journal of Network and Computer Applications*, 192:103164, 2021.

[37] Dragos Mocrii, Yuxiang Chen, and Petr Musilek. Iot-based smart homes: A review of system architecture, software, communications, privacy and security. *Internet of Things*, 1:81–98, 2018.

[38] Sarah Shaharuddin, Khairul Nizam Abdul Maulud, Syed Ahmad Fadhli Syed Abdul Rahman, Adi Irfan Che Ani, and Biswajeet Pradhan. The role of iot sensor in smart building context for indoor fire hazard scenario: A systematic review of interdisciplinary articles. *Internet of Things*, 22:100803, 2023.

[39] Muhammad Shoaib Farooq, Shamyla Riaz, Adnan Abid, Kamran Abid, and Muhammad Azhar Naeem. A survey on the role of iot in agriculture for the implementation of smart farming. *Ieee Access*, 7:156237–156271, 2019.

[40] T Lakshmi Narayana, C Venkatesh, Ajmeera Kiran, Adarsh Kumar, Surbhi Bhatia Khan, Ahlam Almusharraf, Mohammad Tabrez Quasim, et al. Advances in real time smart monitoring of environmental parameters using iot and sensors. *Heliyon*, 10(7), 2024.

[41] G. Sucharitha, Bodepu Tannmayee, and Kanagala Dwarakamai. *Revolution in IoT: Smart Wearable Technology*, pages 407–425. Springer International Publishing, Cham, 2022.

[42] F John Dian, Reza Vahidnia, and Alireza Rahmati. Wearables and the internet of things (iot), applications, opportunities, and challenges: A survey. *IEEE access*, 8:69200–69211, 2020.

[43] Nhu-Ngoc Dao. Internet of wearable things: Advancements and benefits from 6g technologies. *Future Generation Computer Systems*, 138:172–184, 2023.

[44] Afzal Badshah, Anwar Ghani, Ali Daud, Ateeqa Jalal, Muhammad Bilal, and Jon Crowcroft. Towards smart education through the internet of things: A review. *arXiv preprint arXiv:2304.12851*, 2023.

[45] Afzal Badshah, Moustafa M Nasralla, Ateeqa Jalal, and Haleem Farman. Smart education in smart cities: Challenges and solution. pages 01–08, 2023.

[46] SM Abu Adnan Abir, Adnan Anwar, Jinho Choi, and ASM Kayes. Iot-enabled smart energy grid: Applications and challenges. *IEEE access*, 9:50961–50981, 2021.

[47] Sharu Bansal and Dilip Kumar. Iot ecosystem: A survey on devices, gateways, operating systems, middleware and communication. *International Journal of Wireless Information Networks*, 27:340–364, 2020.

[48] Bahareh Pahlevanzadeh, Sara Koleini, and Suzi Iryanti Fadilah. Security in iot: Threats and vulnerabilities, layered architecture, encryption mechanisms, challenges and solutions. In *International Conference on Advances in Cyber Security*, pages 267–283. Springer, 2020.

[49] Saniya Zahoor and Roohie Naaz Mir. Constraints in pervasive iot applications: An analysis. In *Applications of Advanced Computing in Systems: Proceedings of International Conference on Advances in Systems, Control and Computing*, pages 299–307. Springer, 2021.

[50] Deepu Job and Varghese Paul. Challenges, security mechanisms, and research areas in iot and iiot. *Internet of Things and Its Applications*, pages 523–538, 2022.

[51] Poornima M Chanal and Mahabaleshwar S Kakkasageri. Security and privacy in iot: a survey. *Wireless Personal Communications*, 115(2):1667–1693, 2020.

[52] Yuchen Yang, Longfei Wu, Guisheng Yin, Lijie Li, and Hongbin Zhao. A survey on security and privacy issues in internet-of-things. *IEEE Internet of things Journal*, 4(5):1250–1258, 2017.

[53] Rahma Trabelsi, Ghofrane Fersi, and Mohamed Jmaiel. Access control in internet of things: A survey. *Computers & Security*, page 103472, 2023.

[54] Regel Gonzalez-Usach, Diana Yacchirema, Matilde Julian, and Carlos E Palau. Interoperability in iot. *Handbook of Research on Big Data and the IoT*, page 149, 2019.

[55] Sachin Kumar, Pawan Kumar Verma, Rajesh Verma, Maazen Alsabaan, and Tamer Abdelkader. Internet of things: Classification, challenges, and their solutions. In *Applications of Computational Intelligence Techniques in Communications*, pages 137–172. CRC Press, 2024.

[56] Sana Belguith, Nesrine Kaaniche, Abderrazak Jemai, Maryline Laurent, and Rabah Attia. Pabac: a privacy preserving attribute based framework for fine grained access control in clouds. In *SECRYPT 2016: 13th International Conference on Security and Cryptography*, volume 4, pages 133–146. Scitepress, 2016.

[57] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE symposium on security and privacy (SP'07)*. IEEE, 2007.

[58] Dijiang Huang, Qiuxiang Dong, and Yan Zhu. *Attribute-based encryption and access control*. CRC Press, 2020.

[59] Keerti Naregal and Vijay Kalmani. Study of lightweight abe for cloud based iot. In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pages 134–137. IEEE, 2020.

[60] Mohammad Ali Jabraeil Jamali, Bahareh Bahrami, Arash Heidari, Parisa Allahverdizadeh, Farhad Norouzi, Mohammad Ali Jabraeil Jamali, Bahareh Bahrami, Arash Heidari, Parisa Allahverdizadeh, and Farhad Norouzi. Some cases of smart use of the iot. *Towards the Internet of Things: Architectures, Security, and Applications*, pages 85–129, 2020.

[61] Bongsug Kevin Chae. The evolution of the internet of things (iot): A computational text analysis. *Telecommunications Policy*, 43(10):101848, 2019.

[62] Ajay Prasad, Prachi Kapoor, and Thipendra P Singh. Security threats in iot and their prevention. In *Communication Technologies and Security Challenges in IoT: Present and Future*, pages 131–146. Springer, 2024.

[63] Minhaj Ahmad Khan and Khaled Salah. Iot security: Review, blockchain solutions, and open challenges. *Future generation computer systems*, 82:395–411, 2018.

[64] Shivani Agarwal, Sandhya Makkar, and Duc-Tan Tran. *Privacy vulnerabilities and data security challenges in the IoT*. Taylor Francis Group., 2020.

[65] Hokeun Kim and Edward A Lee. Authentication and authorization for the internet of things. *IT Professional*, 19(5):27–33, 2017.

[66] Panagiotis I Radoglou Grammatikis, Panagiotis G Sarigiannidis, and Ioannis D Moscholios. Securing the internet of things: Challenges, threats and solutions. *Internet of Things*, 5:41–70, 2019.

[67] Slavko J Pokorni. Reliability and availability of the internet of things. *Vojnotehnicki glasnik/Military Technical Courier*, 67(3):588–600, 2019.

[68] Saurabh Singh, Pradip Kumar Sharma, Seo Yeon Moon, and Jong Hyuk Park. Advanced lightweight encryption algorithms for iot devices: survey, challenges and solutions. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–18, 2017.

[69] P Prakasam, TR Suresh Kumar, T Velmurugan, and S Nandakumar. Efficient power distribution model for iot nodes driven by energy harvested from low power ambient rf signal. *Microelectronics Journal*, 95:104665, 2020.

[70] P Prakasam, Muthusamy Madheswaran, KP Sujith, and Md Shohel Sayeed. An enhanced energy efficient lightweight cryptography method for various iot devices. *ICT Express*, 7(4):487–492, 2021.

[71] Wade Trappe, Richard Howard, and Robert S Moore. Low-energy security: Limits and opportunities in the internet of things. *IEEE Security & Privacy*, 13(1):14–21, 2015.

[72] Aafaf Ouaddah, Hajar Mousannif, Anas Abou Elkalam, and Abdellah Ait Ouahman. Access control in the internet of things: Big challenges and new opportunities. *Computer Networks*, 112:237–262, 2017.

[73] Jing Qiu, Zhihong Tian, Chunlai Du, Qi Zuo, Shen Su, and Binxing Fang. A survey on access control in the age of internet of things. *IEEE Internet of Things Journal*, 7(6):4682–4696, 2020.

[74] Djamel Eddine Kouicem, Abdelmadjid Bouabdallah, and Hicham Lakhlef. Internet of things security: A top-down survey. *Computer Networks*, 141:199–221, 2018.

[75] Sowmya Ravidas, Alexios Lekidis, Federica Paci, and Nicola Zannone. Access control in internet-of-things: A survey. *Journal of Network and Computer Applications*, 144:79–101, 2019.

[76] Kaushik Ragothaman, Yong Wang, Bhaskar Rimal, and Mark Lawrence. Access control for iot: A survey of existing research, dynamic policies and future directions. *Sensors*, 23(4):1805, 2023.

[77] Ravi S Sandhu and Pierangela Samarati. Access control: principle and practice. *IEEE communications magazine*, 32(9):40–48, 1994.

[78] Ravi S Sandhu. Role-based access control. In *Advances in computers*, volume 46, pages 237–286. Elsevier, 1998.

[79] Anas Abou El Kalam, R El Baida, Philippe Balbiani, Salem Benferhat, Frédéric Cuppens, Yves Deswarte, Alexandre Miege, Claire Saurel, and Gilles Trouessin. Organization based access control. In *Proceedings POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, pages 120–131. IEEE, 2003.

[80] Vincent C Hu, David Ferraiolo, Rick Kuhn, Arthur R Friedman, Alan J Lang, Margaret M Cogdell, Adam Schnitzer, Kenneth Sandlin, Robert Miller, Karen Scarfone, et al. Guide to attribute based access control (abac) definition and considerations (draft). *NIST special publication*, 800(162):1–54, 2013.

[81] Eric Yuan and Jin Tong. Attributed based access control (abac) for web services. In *IEEE International Conference on Web Services (ICWS'05)*. IEEE, 2005.

[82] Jaehong Park and Ravi Sandhu. The uconabc usage control model. *ACM transactions on information and system security (TISSEC)*, 7(1):128–174, 2004.

[83] Yanli Ren, Shuozhong Wang, Xinpeng Zhang, and Zhenxing Qian. Fully secure ciphertext-policy attribute-based encryption with constant size ciphertext. In *2011 Third International Conference on Multimedia Information Networking and Security*, pages 380–384. IEEE, 2011.

[84] Susan Hohenberger and Brent Waters. Online/offline attribute-based encryption. In *Public-Key Cryptography–PKC 2014: 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings 17*, pages 293–310. Springer, 2014.

[85] Shashank Agrawal and Melissa Chase. Fame: Fast attribute-based message encryption. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 665–682, 2017.

[86] Mohammad Bany Taha, Chamseddine Talhi, and Hakima Ould-Slimane. Performance evaluation of cp-abe schemes under constrained devices. *Procedia Computer Science*, 155:425–432, 2019.

[87] Yu Jin, Chuan Tian, Heng He, and Fan Wang. A secure and lightweight data access control scheme for mobile cloud computing. In *2015 IEEE Fifth International Conference on Big Data and Cloud Computing*, pages 172–179. IEEE, 2015.

[88] Mahdi Mahdavi, Mohammad Hesam Tadayon, Mohammad Sayyad Haghighi, and Zahra Ahmadian. Iot-friendly, pre-computed and outsourced attribute based encryption. *Future Generation Computer Systems*, 2023.

[89] Xuanxia Yao, Zhi Chen, and Ye Tian. A lightweight attribute-based encryption scheme for the internet of things. *Future Generation Computer Systems*, 49:104–112, 2015.

[90] Bhavya Alankar, Gaurav Sharma, Harleen Kaur, Raul Valverde, and Victor Chang. Experimental setup for investigating the efficient load balancing algorithms on virtual cloud. *Sensors*, 20(24):7342, 2020.

[91] Muhammed Enes Bayrakdar. Priority based health data monitoring with ieee 802.11 af technology in wireless medical sensor networks. *Medical & Biological Engineering & Computing*, 57(12):2757–2769, 2019.

[92] Lutful Karim, Nidal Nasser, Tarik Taleb, and Abdullah Alqallaf. An efficient priority packet scheduling algorithm for wireless sensor network. In *2012 IEEE international conference on communications (ICC)*, pages 334–338. IEEE, 2012.

[93] Benedetto Girgenti, Pericle Perazzo, Carlo Vallati, Francesca Righetti, Gianluca Dini, and Giuseppe Anastasi. On the feasibility of attribute-based encryption on constrained iot devices for smart systems. In *2019 IEEE international conference on smart computing (SMARTCOMP)*, pages 225–232. IEEE, 2019.

[94] Shafi Ullah, Raja Zahilah Radzi, Tulha Moaiz Yazdani, Ali Alshehri, and Ilyas Khan. Types of lightweight cryptographies in current developments for resource constrained machine type communication devices: Challenges and opportunities. *IEEE Access*, 10:35589–35604, 2022.

[95] Wafa'a Kassab and Khalid A Darabkh. A–z survey of internet of things: Architectures, protocols, applications, recent advances, future directions and recommendations. *Journal of Network and Computer Applications*, 163:102663, 2020.

[96] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer networks*, 52(12):2292–2330, 2008.

[97] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, and Andrea Passarella. Energy conservation in wireless sensor networks: A survey. *Ad hoc networks*, 7(3):537–568, 2009.

[98] Amit Sinha and Anantho Chandrakasan. Dynamic power management in wireless sensor networks. *IEEE Design & Test of Computers*, 18(2):62–74, 2001.

[99] Djawhara Benchaira, Okba Tibermacine, Walid Bechkit, and Abdelmalik Bachir. Leveraging predictability for global optimization of iot networks. In *ICC 2022-IEEE International Conference on Communications*, pages 3586–3591. IEEE, 2022.