People's democratic republic of algeria

Ministry of higher education and scientific research

University of abbes laghrour khenchela

Faculty of sciences and technology

Departement of computer science

Thesis of the degree of Doctorat(LMD)

Speciality : security and technology web

Authentication protocols in internet of things

presented by

Zerraza Imane

Presentation date : February 5,2026

Jury Members

Meriem DJEZZAR	MCA-University of Khenchela	President
Zianou AHMED SEGHIR	Prof-University of Khenchela	Director
Mounir HEMAM	Prof-University of Khenchela	Co-Director
Abdelaali BEKHOUCHE	MCA-University of Khenchela	Examiner
Fella HACHOUF	Prof-University of constantine1	Examiner
Mohamed Ali BOUANAKA	MCA-University of constantine2	Examiner

2025/2026

Abstract

The Internet of Things (IoT) is revolutionizing the field of information technology, establishing itself across various sectors, from smartwatches to automated factories. It simplifies daily life and provides significant value to both individuals and businesses. However, the diversity of connected devices, which utilize different technologies and have limited resources, presents major security challenges. This thesis aims to design new secure protocols to ensure the authentication of connected objects, addressing the lack of solutions adapted to the diverse standards, infrastructures, and platforms within the IoT ecosystem. This thesis aims to design an innovative authentication and access control mechanism specifically tailored for IoT edge environments. The proposed approach is intended to support networks composed of a large number of nodes, optimizing data transmission under bandwidth constraints by leveraging the lightweight ChaCha20 symmetric encryption algorithm for secure session key establishment. A key objective is to enhance identity management and access control by integrating the HoBAC framework with blockchain technology, thereby promoting greater precision, scalability, and trust across distributed edge nodes. In parallel, the thesis seeks to maintain a unified and efficient encryption strategy throughout the edge network by consistently using the ChaCha20 algorithm for session key generation. The overall goal is to develop an authentication scheme that ensures mutual authentication and secure communication among key IoT components, including sensor nodes, remote users, and edge servers.

Keywords : the Internet of Things (IoT), Security,Cryptographic algorithmes, Authentification.

الملخص

يحدث انترنت الاشياء (IOT) ثورة في مجال تكنولوجيا المعلومات , حيث يرسخ مكانته في مختلف القطاعات,من الساعات الذكية الى المصانع المؤتمنة. فهو يبسط الحياة اليومية و يوفر قيمة كبيرة لكل من الافراد والشركات. و مع ذلك,فان تنوع الاجهزة المتصلة, التي تستخدم تقنيات مختلفة وتتمتع بموارد محدودة, يطرح تحديات كبيرة في مجال الامن السيبراني. تهدف هذه الرسالة الى تصميم بروتوكولات امنة جديدة لضمان مصادقة الاجهزة المتصلة, من خلال معالجة النقص في الحلول الملائمة للتنوع في المعايير و بنى التحتية, و المنصات داخل نظام انترنت الاشياء.

تهدف هذه الرسالة إلى تصميم آلية مبتكرة للمصادقة والتحكم في الوصول, مخصصة بشكل خاص لبيئات الحوسبة الطرفية في إنترنت أألشياء. يقترح ان تدعم هذه اآللية الشبكات التي تتكوّن من عدد كبير من العقد، مع تحسين نقل البيانات ضمن قيود النطاق الترددي، من خالل االستفادة من خوارزمية التشفير المتماثل الخفيفة Chacha20 إلنشاء مفاتيح جلسات آمنة. ويتمثل أحد أألهداف الرئيسية في تعزيز إدارة الهوية والتحكم في الوصول من خالل دمج HoBAC مع تقنية البلوكشين، مما يعزز الدقة، وقابلية التوسع، والثقة بين العقد الطرفية الموزعة. بالتوازي مع ذلك، تسعى الرسالة إلى الحفاظ على استراتيجية تشفير موحدة وفعالة عبر شبكة الحافة، من خالل االستخدام المتسق لخوارزمية Chacha20 في توليد مفاتيح الجلسات. الهدف النهائي هو تطوير مخطط مصادقة يضمن المصادقة المتبادلة و التواصل اآلمن بين المكونات الرئيسية إلنترنت أألشياء، بما في ذلك عقد االستشعار، والمستخدمين عن بُعد، وخوادم الحافة.

الكلمات المفتاحية:

انترنت الاشياء (IOT) , الامن السيبراني, خوارزميات التشفير' التشفير

Résumé

L'Internet des Objets (IoT) révolutionne le domaine des technologies de l'information en s'imposant dans divers secteurs, allant des montres connectées aux usines automatisées. Il simplifie la vie quotidienne et apporte une valeur significative tant aux individus qu'aux entreprises. Toutefois, la diversité des objets connectés, qui utilisent des technologies variées et disposent de ressources limitées, soulève d'importants défis en matière de sécurité. Cette thèse vise à concevoir de nouveaux protocoles sécurisés afin de garantir l'authentification des objets connectés, en répondant au manque de solutions adaptées à la diversité des standards, des infrastructures et des plateformes de l'écosystème IoT. Elle propose la conception d'un mécanisme innovant d'authentification et de contrôle d'accès, spécifiquement adapté aux environnements edge de l'IoT. L'approche envisagée est conçue pour prendre en charge des réseaux composés d'un grand nombre de nœuds, tout en optimisant la transmission des données sous des contraintes de bande passante, grâce à l'utilisation de l'algorithme de chiffrement symétrique léger ChaCha20 pour l'établissement sécurisé de clés de session. Un objectif essentiel est d'améliorer la gestion des identités et le contrôle d'accès en intégrant le model HoBAC avec la technologie blockchain, afin de renforcer la précision, la scalabilité et la confiance entre les nœuds distribués. Parallèlement, la thèse vise à maintenir une stratégie de chiffrement unifiée et efficace au sein du réseau edge, en utilisant de manière cohérente l'algorithme ChaCha20 pour la génération des clés de session. L'objectif global est de développer un schéma d'authentification assurant une authentification mutuelle et une communication sécurisée entre les composants clés de l'IoT, notamment les capteurs, les utilisateurs distants et les serveurs edge.

Mots-clés : Internet des Objets (IoT), Sécurité, Algorithmes cryptographiques, Authentification

Table of Contents

List of Figures

List of Tables

List of Abbreviations

IoT	Internet of Things
WSN	Wireless Sensor Network
ECC	Elliptic Curve Cryptography
IdM	Identity Management System
HoBAC	Higher Order Attribute-Based Access Control
PoW	Proof of Work
PoS	Proof of Stake

General introduction

Research Context and Problem Statement The Internet of Things (IoT) has rapidly evolved into a foundational technology driving innovation across numerous sectors such as healthcare, smart cities, industrial systems, and intelligent transportation. By enabling continuous data exchange and interaction between devices, people, and their environments, IoT systems promise enhanced efficiency, automation, and real-time decision-making. However, this expansive interconnectivity introduces significant vulnerabilities, especially concerning data security and user privacy. Most IoT devices are resource-constrained, lacking the computational power and memory needed to support conventional security mechanisms. This makes them particularly susceptible to cyber threats, unauthorized access, and identity spoofing. The challenge lies in implementing robust security solutions that not only protect user data but also accommodate the limited capabilities of IoT devices. As a result, the development of secure, lightweight authentication mechanisms remains an urgent and open research problem.

Research Objectives

The primary objective of this research is to address the security limitations of IoT ecosystems by designing and evaluating lightweight authentication protocols tailored for constrained devices. This includes investigating existing threats to identity management and device authentication within IoT networks, understanding the limitations posed by hardware and power restrictions, and identifying the trade-offs between performance and security. The research aims to propose a novel authentication framework that ensures mutual trust between devices while minimizing computational overhead. In doing so, the study seeks to improve the resilience of IoT systems to common attacks such as impersonation, replay, and man-in-the middle, thereby fostering greater confidence in IoT deployments across sensitive and large scale applications.

Research Contributions

This research contributes to the field of IoT security by introducing a context-aware, lightweight authentication protocol specifically optimized for the unique constraints of IoT environments. Unlike traditional security models, the proposed solution emphasizes low-latency communication, minimal consumption, and adaptability to heterogeneous IoT devices.

- An Efficient Lightweight Authentication and Access Control for IOT Edge Devices : an innovative authentication and access control mechanism specifically designed for IoT edge environments. The approach is optimized for networks composed of a large number of nodes and supports efficient data transmission under bandwidth constraints by leveraging the lightweight ChaCha20 symmetric encryption algorithm for secure session key establishment. Furthermore, it strengthens

identity management and access control by incorporating the HoBAC framework in combination with blockchain technology, ensuring greater precision, scalability, and trust across distributed edge nodes.

- Lightweight Authentication for IOT Edge Devices : The scheme consistently utilizes the ChaCha20 symmetric encryption algorithm to generate session keys, providing a unified and lightweight approach to securing communications between entities in the edge network. The proposed authentication scheme aims to provide mutual authentication and establish secure communication between a sensor node, a remote user, and the edge server.

Thesis organization

This thesis is structured into several chapters, with the two chapters forming the context and state of the art part. Each addressing a critical aspect of IoT security and authentication.

Chapter 1 : Concepts, technologies, and challenges in the internet of things

The first chapter lays the groundwork for the thesis by providing a broad and structured introduction to the Internet of Things. It begins with definitions and visions of IoT, highlighting how the concept has evolved and the impact it is expected to have across various sectors. Next, it discusses the principal technologies that enable IoT systems, including sensing devices, wireless communication protocols, and edge computing. The chapter then explores the application domains of IoT, such as smart homes and smart cities. Following this, it addresses the problematics and challenges inherent in IoT environments, focusing on issues like architecture, security, and optimization metrics. Finally, the chapter introduces cryptography basics, presenting fundamental concepts necessary for understanding and designing secure communication in IoT systems.

Chapter 2 : Authentication and Identity Management in the Internet of Things

The second chapter delves into the critical aspects of authentication and identity management within the IoT context. It begins by discussing the principle of identity in logical philosophy, offering a conceptual foundation for understanding digital identity. The chapter then outlines the requirements of identity management systems, emphasizing essential factors such as scalability, interoperability, and mobility, which are crucial for effective identity management in dynamic and heterogeneous IoT environments. A review of classic identity management systems on the Internet is provided to highlight traditional approaches and their limitations when applied to IoT. The chapter further introduces blockchain technology as a promising alternative, beginning with an overview of blockchain, followed by an explanation of its architecture, key characteristics, and the basic mechanism of how blockchain works. It also covers various consensus algorithms and existing blockchain platforms that support decentralized identity management. The section then explores blockchain-based identity management systems (IdMS), highlighting their potential to enhance trust, transparency, and security. The chapter concludes with a detailed examination of authentication in the Internet of Things, including a summary and analysis of existing

authentication schemes, setting the stage for the thesis's proposed solutions.

Chapter 3 : Contribution and proposed solutions

This chapter presents the original contributions developed in this thesis to address the challenges of secure communication and identity management in IoT edge environments. The first contribution, titled "An Efficient Lightweight Authentication and Access Control for IoT Edge Devices," introduces a novel authentication scheme tailored for large-scale and resource constrained networks. The section begins with a detailed explanation of the proposed authentication scheme, which leverages lightweight cryptographic techniques to establish secure communication channels. It integrates a Higher Order Attribute-Based Access Control (HoBAC) model, enhancing access precision and scalability across distributed nodes. The chapter continues with the implementation of this solution, followed by a comprehensive security analysis of the proposed protocol, highlighting its resistance to known attacks. A performance evaluation is conducted to validate the protocol's efficiency in terms of computational cost, communication overhead, and energy consumption. The second contribution, titled "Lightweight Authentication for IoT Edge Devices," offers a simplified authentication solution designed for rapid deployment in IoT edge scenarios. This section outlines the proposed authentication scheme, focusing on mutual authentication and secure session establishment between sensor nodes, users, and edge servers. A security analysis of the proposed protocol confirms its robustness against typical IoT threats, and a performance evaluation demonstrates its lightweight nature and suitability for devices with limited resources.

Context

and state of

art

Concepts, technologies, and challenges in the internet of things

1 Introduction

The rapid development of digital technologies has given rise to new paradigms that are reshaping how we interact with our environment. Among these, the Internet of Things (IoT) stands out as a major technological revolution that is enabling everyday objects to communicate, collect, and exchange data. This chapter lays the foundation for understanding the IoT and the technical ecosystem it relies on. In this chapter, we begin by clearly defining what the Internet of Things (IoT) is ? and examining the technologies used in this new paradigm of wireless network communications. Next, we present the various application domains of IoT, demonstrating its widespread impact across sectors such as smart home and smart cities. We then highlight the different challenges and issues that need to be addressed in the IoT field, including security and scalability, all of which are critical to the successful deployment and operation of IoT systems.

2 Defintions and IOT technologies

In this section, we define the IoT, the different perspectives, and the various technologies used in this context.

2.1 Definitions and visions

The Internet of Things (IoT) can be encapsulated as the interconnection of sensing and actuating devices. This interconnectedness empowers the seamless sharing of information across diverse platforms through a unified framework, Promoting the establishment of a shared operational perspective. This, in turn, Enables the implementation of inventive applications through seamless extensive sensing, advanced data

analytics, and efficient information representation. The combination of state-of-the-art pervasive sensing and cloud computing technologies is essential for reaching these aims [1][2]. IOT allows each element, whether currently linked or slated for connection, must be distinguished by its unique identification, location, and functionalities.[1]

The emerging Internet of Things is set to influence various application domains. These applications can be categorized according to factors such as network availability, coverage, scale, heterogeneity, repeatability, user involvement, and impact. [1]

Ubiquitous Computing (ubicomp) [3] " the physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network ". The development of the Internet has represented a significant milestone in realizing ubicomp's vision, facilitating communication between individual devices across the globe [1]

For technology to seamlessly integrate into users' daily lives, the Internet of Things requires: (1) a unified perception of the environment, including user interactions and device activities, (2) robust software frameworks and pervasive communication networks that efficiently manage and transmit contextual data to relevant destinations, and (3) advanced analytical tools designed to enable intelligent and autonomous system behavior. By establishing these key components, the IoT can achieve smart connectivity and adaptive computing capabilities. [1]

2.2 The principales technologies

The deployment of IoT in our routine activities is feasible through the amalgamation of multiple technologies. We will give a brief on the different technologies and the distinct roles they will fulfill in IoT.

2.2.1 Identification, detection, and communication technologies

The Internet of Things (IoT) represents a major technological evolution that transforms the way we interact with our environment, by increasingly integrating smart devices into our daily lives. [1]

As IoT deployments continue to expand in scale often involving thousands or even millions of interconnected sensors and devices ,efficient communication mechanisms are essential to maintain seamless data flow and coordination across the network. Edge computing helps address these challenges by decentralizing data processing, enhancing scalability, and supporting reliable communication in both urban and remote environments. This approach not only improves performance but also strengthens system resilience, privacy, and security, making it a cornerstone for large-scale IoT applications.[1][2]

The unique identification of devices is essential for the successful implementation of the Internet of Things (IoT). Each device must have a distinct identifier, location, and functionality to enable seamless

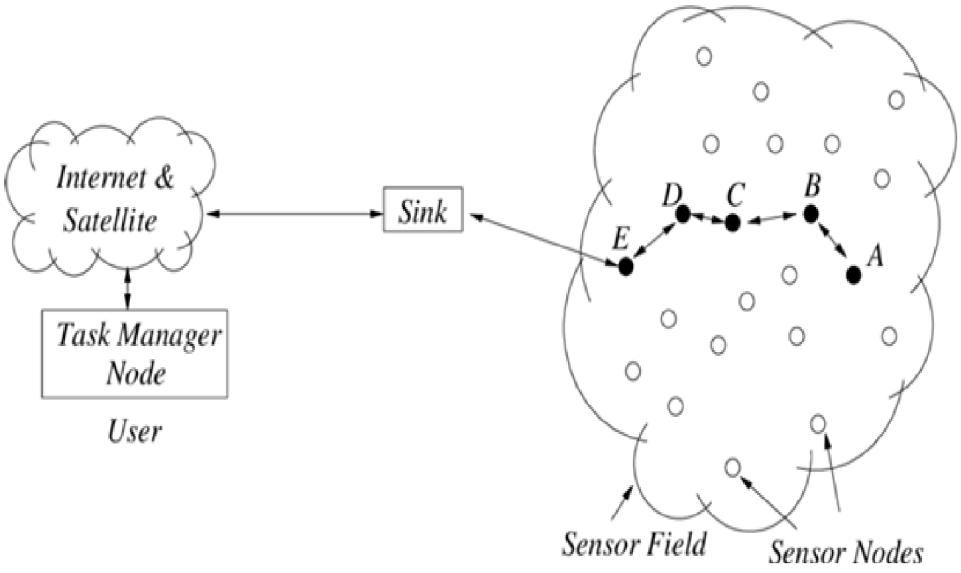

Figure 1.1: Sensor nodes deployed throughout the sensor field[4]

remote control and data exchange. While IPv4 allows geographical identification of sensor groups, it lacks the capability to individually address devices. IPv6 improves mobility and identification but still faces challenges due to the diverse nature of IoT devices and data flows. The Uniform Resource Name (URN) system plays a key role in addressing these issues by creating accessible replicas of resources. Additionally, lightweight IPv6 solutions are being developed to enable unique addressing of home appliances. Wireless sensor networks, which form the foundation of IoT, require a subnet with a gateway that utilizes URNs for device addressing. By integrating these addressing mechanisms, IoT can establish a scalable, reliable, and efficient network for seamless connectivity and control. [1]

sensor nodes are typically distributed across a sensor field(figure1.1), where they collect and transmit data. Using a multihop infrastructureless architecture, the data is routed through a sink to reach the end user. The sink serves as an intermediary, facilitating communication with the task manager node via the Internet or satellite, ensuring efficient data transmission and network connectivity. [4]

SENSEI[5] is a pioneering European project aimed at realizing the Real World Internet (RWI) by integrating heterogeneous wireless sensor and actuator networks (WSANs) into a global framework. With a large-scale consortium of industrial leaders, research institutions, and universities, SENSEI focuses on scalability, interoperability, security, and real-time data access to support various applications, including smart cities, emergency management, and industrial automation. The project employs service-oriented architecture (SOA) and Semantic Web technologies to ensure seamless interaction between physical and digital environments. Ultimately, SENSEI strives to create a secure, efficient, and business-driven infrastructure, fostering an open marketplace for context-aware services and real-world interactions.

Recent progress in low-power circuits and wireless communications have brought forth cost effective and energy-efficient miniature devices suitable for deployment in remote sensing applications. The convergence of these technological breakthroughs enhances the feasibility of deploying a Wireless Sensor Network (WSN) comprised of numerous intelligent sensors. This network facilitates the collection, processing, analysis, and dissemination of valuable information in diverse environments [4]

IEEE802.15.4 :The protocol offers flexibility through various topologies (e.g., mesh, star) and benefits from low memory requirements, making it suitable for resource-constrained environments. Integration with protocols like 6LoWPAN allows for IPv6 compatibility, extending local networks to the global Internet.[2] .

ZigBee[6], built upon the IEEE 802.15.4 standard, offers a low-cost, low-power, and low-range solution ideal for Personal Area Networks (PANs) in smart city applications. With features like robust routing (via AODV), AES-128 encryption for security, and support for up to 65,000 nodes in various topologies (star, mesh, tree), ZigBee ensures reliability, interoperability, and mobility. Its maturity and extensive use in domains like smart grids, transportation, healthcare, and environmental monitoring highlight its widespread applicability.[2]

Low Power Wide Area Networks (LPWANs) [7]introduce a transformative communication model that complements existing cellular and short-range wireless technologies. By enabling wide-area connectivity with low power consumption and support for low data rates, LPWA technologies effectively address the unique and diverse requirements of IoT applications that traditional wireless solutions cannot meet. LPWAN aim to bridge the gap between wide area coverage of cellular networks and low power consumption of WPANs and WLANs.[2]

LoRaWAN[8] is a cost-effective, long-range, and low-power communication technology that excels in wide-area, low-data-rate applications such as smart grids and smart metering. Its ability to support tens of thousands of end devices per gateway makes it well-suited for outdoor deployments. Additionally, its publicly available specifications and independence from third party operators facilitate deployment and adoption. [2]

Sigfox[9] is a low-power, long-range communication technology that leverages Ultra Narrow Band (UNB) to achieve wide coverage and high energy efficiency at low deployment costs. Its client-server architecture eliminates synchronization overhead, enhancing scalability and robustness against noise. However, its limited data rate, strict regulatory constraints, and scarce downlink opportunities significantly impact its usability for applications requiring frequentupdates or real-time control. Additionally, lack of encryption mechanisms makes security a concern, as it mainly relies on frequency hopping and an unpredictable payload format for data protection. [2]

2.2.2 Middleware

A method to integrate cyber infrastructure with a Service Oriented Architecture (SOA) and sensor networks to enable access to diverse sensor resources in a deployment-agnostic manner. [1]

The OSWA (OPEN SENSORWEB)is an implementation of the SWE method. The diverse elements described for OSWA are outlined in Figure (1.2). We can distinguish four fundamental layers: Fabric, Services, Development and Application.[10]

The Sensor Fabric layer incorporates the Operating System and application software deployed on physical sensors, enabling them to record observations and commssunicate with each other. At the moment, the programming and deployment of applications at the Fabric layer fall to developers.

A primary goal of the OSWA is to offer a software framework that streamlines the process of application development for heterogeneous wireless sensor networks. After services have been deployed, we can additionally abstract the specifics of the services into interfaces, which we combine to create an API, establishing the foundation of the Application Development layer. Developers can then utilize the API to construct and deploy sensor applications, establish connections between services and construct task management schedules through an interactive GUI.

Fabric Layer: This layer comprises the Operating System and application code installed on physical sensors. It allows sensors to capture observations and communicate among themselves. Developers are responsible for programming and deploying applications at this layer.

Service Layer: This layer provides fundamental services for the OSWA framework. It includes low-level components that offer basic services and higher-level components that provide tools for creating applications and managing the life Lifecycle of data collected via sensor networks. The Service layer is implemented using WSRF (Web Service Resource Framework) services, which allow for persistent relationships between services and enable them to push data out to clients following a publish-subscribe paradigm.

Development Layer: This layer facilitates the development of applications for heterogeneous wireless sensor networks. It abstracts the details of services into interfaces, forming the basis of the Application Development layer. Developers can use APIs (Application Programming Interfaces) provided at this layer to build and Implement sensor solutions, establish connections among services, and create job scheduling frameworks using an interactive graphical user interface.

Application Layer: This layer encompasses the actual sensor applications built using the APIs provided by the Development layer. It allows developers to define specific application logic and functionalities tailored to their requirements.

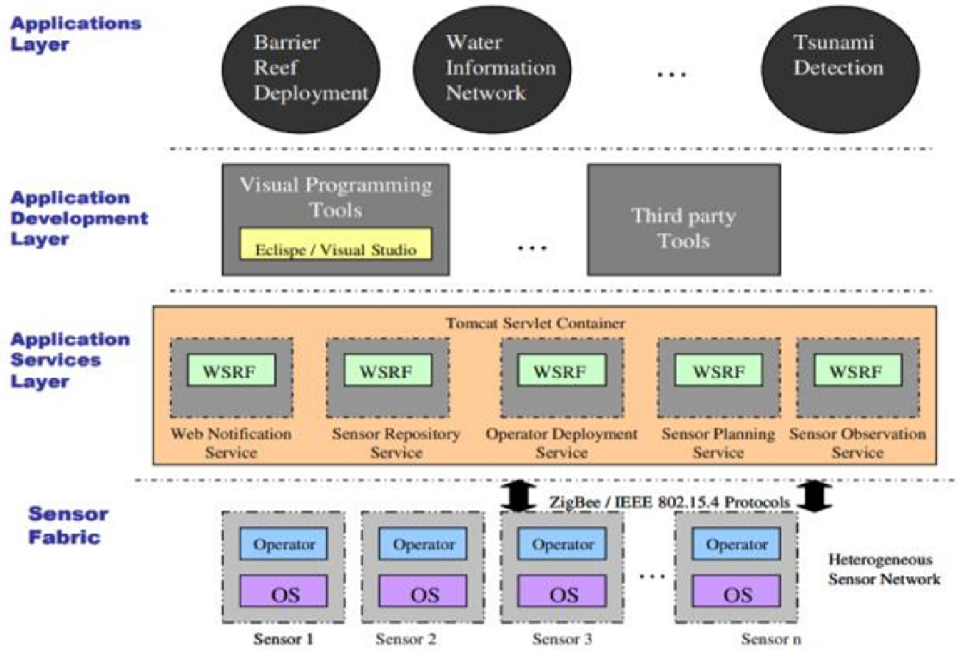

Figure 1.2: High level view of OSWA[10]

3 Application domains of IOT

Today, the Internet of Things (IoT) enables the development of a wide range of applications across various sectors and environments, primarily aimed at enhancing everyday life. These applications rely on smart devices, which often face limitations in communication and energy capacity. Despite these constraints, the devices interact with one another to gather valuable data and deliver meaningful outcomes. IoT applications can be categorized into several distinct fields depending on their use [1] :

- ✓ personal and home
- ✓ Enterprise
- ✓ Utilities
- ✓ Mobile

3.1 personal and home

Personal and Home IoT refers to the deployment of interconnected smart devices and sensor technologies within personal living spaces and private networks. This ecosystem typically includes a wide range of devices such as smartwatches, fitness trackers, voice assistants, smart TVs, connected kitchen appliances,

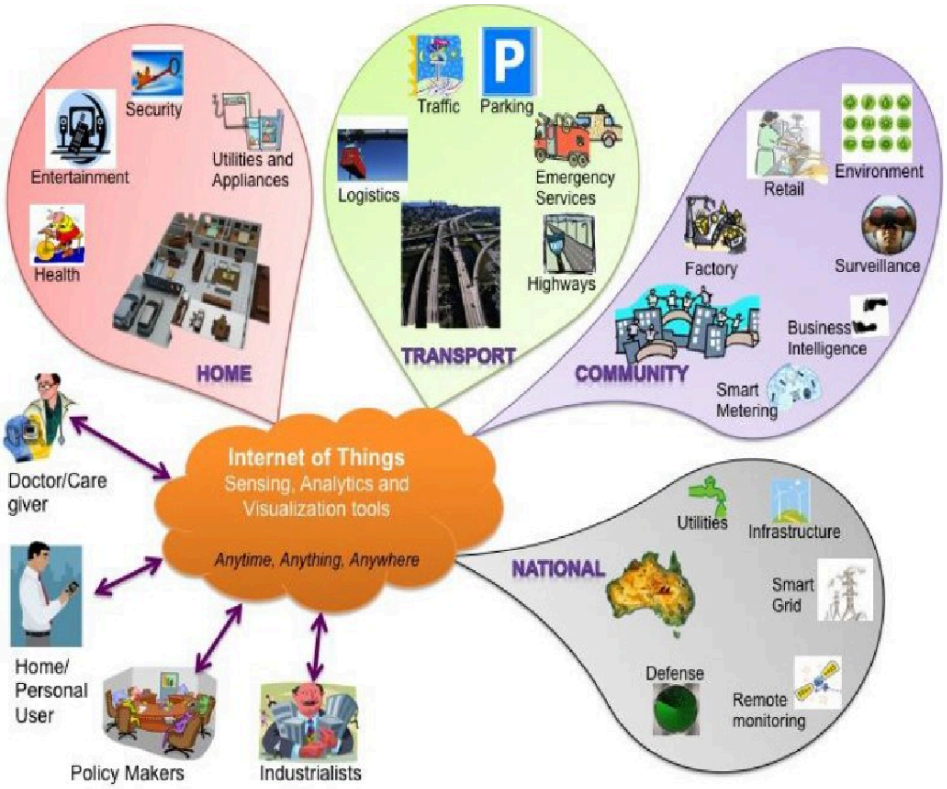

Figure 1.3: application domains[1]

security systems, and environmental sensors. These devices gather and share real-time data to support personalized services and streamline daily routines. For example, wearable health devices[11][12] can monitor heart rate and physical activity, while home automation systems can control lighting, temperature, and security remotely based on user preferences or behavioral patterns. While offering convenience and efficiency, ensuring robust security measures will be crucial for its widespread adoption and success.[1]

3.2 Enterprise

Enterprise IoT refers to the deployment and integration of interconnected devices, sensors, and intelligent systems within a business or industrial setting. These connected technologies are embedded throughout an organization's infrastructure including manufacturing equipment, office environments, supply chains, and logistics system to gather and transmit real-time data relevant to operations, performance, and conditions. This data is then leveraged by business owners, managers, and stakeholders to support informed decision-making, optimize processes, and drive strategic outcomes. However, the integration of Enterprise IoT also introduces challenges, particularly around scalability, interoperability, data management, and cybersecurity.[1]

3.3 Utilities

Utilities in the context of the Internet of Things (IoT) refer to the application of connected technologies to enhance the management, monitoring, and delivery of essential services such as electricity, water, gas, and waste management. The primary objective is to optimize resource use, improve service reliability, and reduce operational costs through real-time data collection, automation, and intelligent analytics. By embedding sensors, meters, and communication devices throughout utility infrastructure, providers gain deeper visibility into system performance and customer usage patterns. [1]

3.4 Mobile

Mobile IoT refers to the deployment of Internet of Things technologies across mobile and dynamic environments, particularly in sectors such as smart transportation, logistics, fleet management, and urban mobility. This domain focuses on enhancing the efficiency, safety, and sustainability of systems that involve the movement of people, goods, and services. It relies on a wide array of connected sensors, GPS devices, wireless communication protocols (such as Bluetooth, LTE-M, and NB-IoT), and advanced data analytics to collect, process, and respond to information in real time. As Mobile IoT systems become more advanced, they are increasingly integrating with technologies such as artificial intelligence, edge computing, and 5G connectivity to support ultra-low latency, massive device networks, and smarter automation. [1]

4 problematics and challenges

IoT systems are built upon a combination of various underlying technologies, and as a result, the challenges associated with IoT can be broadly categorized into two main groups:

1. Challenges stemming from the foundational technologies that IoT depends on.
2. Novel challenges that arise specifically from the deployment and integration of IoT systems in real-world environments.

The main challenges for IoT systems are :

1. Architecture
2. Programmability
3. Optimization metrics
4. Privacy and security

4.1 Architecture

Early architectural choices in IoT research have significantly shaped the field's development, affecting scalability, security, and integration with emerging technologies. Initial efforts focused heavily on wireless sensor networks (WSNs), which, while foundational, faced issues with interoperability and responsiveness. Projects like SENSEI and IoT-A advanced the field by promoting standardization and more flexible architectures[1]. However, current models such as secure IoT frameworks for smart cities still face limitations, particularly with integrating Software-Defined Networking (SDN) and ensuring node security and network efficiency at scale.[13]

4.2 Programmability

The deployment of networks at large scale raises significant concerns regarding security and privacy. Attacks on these networks can range from disabling availability to accessing personal information. Cryptography serves as the primary defense against data corruption, particularly in vulnerable components like RFID, WSN, and cloud systems. While encryption safeguards data confidentiality against outsider attacks, insider threats require additional non cryptographic measures, especially in WSNs. Secure reprogramming protocols are essential for updating sensor applications remotely, ensuring authentication of code updates and preventing malicious installations. Further research is needed to enhance cryptographic methods in network reprogramming protocols. [1]

4.3 Optimization metrics

Heterogeneous networks inherently support multiple services, necessitating the ability to accommodate various types of traffic without compromising Quality of Service (QoS). These networks typically handle both throughput-oriented, delay-tolerant traffic (e.g., weather monitoring) and bandwidth-sensitive, real-time traffic (e.g., traffic monitoring). Moreover, different data-related applications within these traffic classes may have distinct QoS requirements. Achieving QoS guarantees in wireless networks is challenging due to resource allocation limitations in shared wireless media. Quality of Service in cloud computing is also a critical research area, especially with the increasing availability of data and tools on clouds. [1][14]

4.4 Privacy and security

The IoT is extremely vulnerable to various possible attacks such as side channel attacks, replay attacks, identity theft, etc. Indeed, devices can become defective due to physical attacks on the hardware [1] [2]. Furthermore, most communications are wireless, which makes eavesdropping on communication channels easier. Most devices are limited in terms of computational capacity, making the implementation of complex security systems impractical.

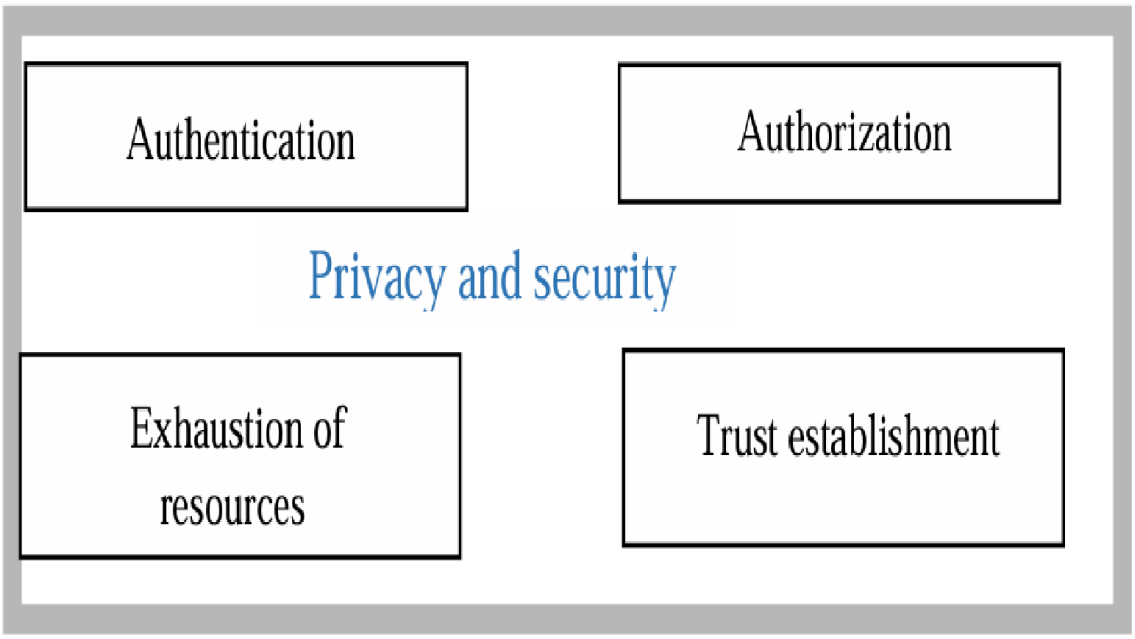

Figure 1.4: Privacy and security in IoT

The main security issues concern authentication, identity management, and data integrity. Among these, authentication and identity management are the most difficult to ensure. In traditional Internet environments, authentication and certification infrastructures are used to manage identities and ensure authentication. However, this is often not the case in resource constrained IoT environments [13]. The various challenges and research directions related to IoT security are illustrated in the following figure(1.4):

In the following, we will clarify the main and most important security properties in IoT, which are currently the subject of extensive research within the scientific community:

- **Authentication :**

  Authentication plays a critical role in enabling seamless integration and secure communication among IoT devices in various smart environments. Techniques such as hashing, element extraction, and Elliptical Curve Cryptography (ECC) are utilized to establish authentication mechanisms within the IoT ecosystem. While these methods enhance security, challenges such as high communication overhead and susceptibility to jamming attacks remain. Efforts to improve authentication protocols include proposals for Role-Based Access Control (RBAC) and identity authentication models, leveraging techniques like timestamp-based verification and lightweight authentication protocols. Despite advancements, there is a need for further research to address authentication issues across different IoT layers comprehensively and mitigate potential security vulnerabilities

effectively. [13]

- **Authorization :**

  Authorization plays a crucial role in ensuring information security and access control, especially in healthcare devices and e-health applications reliant on interconnected nodes. The IoT involves various users, including humans, machines, and internal/external objects, necessitating robust authorization mechanisms to prevent unauthorized access to sensitive data. Managing and controlling data in heterogeneous IoT environments is a critical challenge, requiring clear data management mechanisms and procedures to ensure data protection throughout the process. Approaches such as ID authentication, like the one proposed by Gaur et al, can enhance security by employing dynamic cipher schemes and pre-shared matrices. However, the effectiveness of such approaches depends on secure key management practices and the sensitivity of the IoT deployment context. Continued research and development are essential to implement robust authorization mechanisms across diverse IoT frameworks. [13]

- **Exhaustion of resources :**

  The exhaustion of resources presents a significant challenge in IoT systems, impacting the performance of applications and leading to resource leakages and overloading. Resource depletion attacks, such as routing loops and consistent high-volume packet transmissions, drain the energy of IoT nodes and accelerate battery degradation. These attacks not only render sensor nodes unavailable but also isolate them in sub-networks, exacerbating the severity of the situation. Effective mitigation strategies are essential to safeguard IoT systems against resource exhaustion attacks and ensure their resilience in the face of evolving threats. [13]

- **Trust establishment :**

  Establishing trust within IoT systems is essential for ensuring interoperability among devices and preserving user privacy. Mechanisms for verifying network devices exist, but establishing trust in verifying network applications remains a challenge. Trust mechanisms must address the portability and mobility of IoT devices, enabling smooth movement between owners while maintaining access control and authorization. Effective trust establishment mechanisms are vital for the security and reliability of IoT ecosystems.[13]

5 Cryptography basics

Cryptography encompasses both the art and science of safeguarding messages, such as email communications and credit card details, during transmission over networks by employing encryption algorithms. These algorithms, utilized extensively in network security, are categorized into three fundamental types:

Symmetric algorithms, Asymmetric (or public-key) algorithms, and Cryptographic protocols. Symmetric algorithms, also known as secret key encryption, employ a single key for both encryption and decryption processes. In contrast, asymmetric algorithms utilize two keys: a public key for encryption and a private key for decryption. [15]

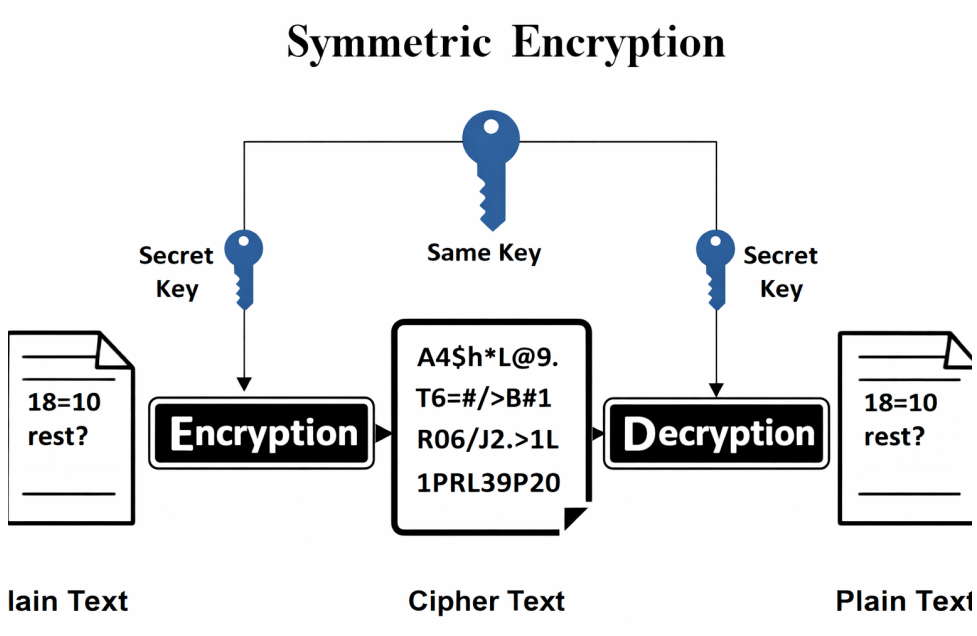

Figure 1.5: Symmetric encryption

5.1 Symmetric key cryptography

In symmetric encryption, the same confidential code is applied to both conceal and reveal data. The communicating individuals must have mutual access to this code to interpret the information. Although this system operates quickly, one major issue is how to safely deliver the key between users. The process relies on identical codes for transforming readable data into a protected format and back again.

5.1.1 Chacha20

ChaCha [16][17]stream encryption methods are engineered to perform efficiently within software frameworks. ChaCha was initially designed with flexible configurations, enabling users to balance speed and protection based on their specific needs , Users can customize ChaCha by adjusting the sizes of the key, nonce, and counter, and by selecting how many rounds the algorithm runs . This part outlines ChaCha20 according to RFC7539, which defines a cautious configuration prioritizing strong security . Assuming 64-byte segments of plaintext and ciphertext are represented as (P_i,C_i), where $P = p_1\|p_2\|\ldots$.and $C = c_1\|c_2\|\ldots$. the ChaCha20 stream cipher ChaCha20-SC : $\{0,1\}^{256} \times \{0,1\}^{96} \times \{0,1\}^* \rightarrow \{0,1\}^*$, $(K, N, P) \mapsto C$ encrypts the plaintext P into a ciphertext C using a 32-byte secret key K and a 12-byte nonce N (figure1.6).

Algorithm 1 ChaCha20 Stream Cipher:
C = ChaCha20-SC(K, N, P).

Input: $K \in \{0,1\}^{256}$, $N \in \{0,1\}^{96}$, $P \in \{0,1\}^{*}$
Output: $C \in \{0,1\}^{|P|}$

for $i \leftarrow 0$ **to** $\lceil |P|/512 \rceil - 1$ **do**
  /* Init State */
  S[0] $\leftarrow$ 0x61707865, S[1] $\leftarrow$ 0x3320646e
  S[2] $\leftarrow$ 0x79622d32, S[3] $\leftarrow$ 0x6b206574
  S[4..11] $\leftarrow K$ {Set Key}
  S[12] $\leftarrow i$ {Set Counter}
  S[13..15] $\leftarrow N$ {Set Nonce}
  S′ $\leftarrow$ S {Save Initial State}
  for $n \leftarrow 0$ **to** 9 **do** {10 Double Rounds}
    /* Column Round */
    S[0, 4, 8, 12] $\leftarrow$ QR(S[0], S[4], S[8], S[12])
    S[1, 5, 9, 13] $\leftarrow$ QR(S[1], S[5], S[9], S[13])
    S[2, 6, 10, 14] $\leftarrow$ QR(S[2], S[6], S[10], S[14])
    S[3, 7, 11, 15] $\leftarrow$ QR(S[3], S[7], S[11], S[15])
    /* Diagonal Round */
    S[0, 5, 10, 15] $\leftarrow$ QR(S[0], S[5], S[10], S[15])
    S[1, 6, 11, 12] $\leftarrow$ QR(S[1], S[6], S[11], S[12])
    S[2, 7, 8, 13] $\leftarrow$ QR(S[2], S[7], S[8], S[13])
    S[3, 4, 9, 14] $\leftarrow$ QR(S[3], S[4], S[9], S[14])
  end for
  $k_i^N \leftarrow$ S $\boxplus$ S′ $\{\forall 0 \leq x \leq 15 : \text{S}[x] \boxplus \text{S}'[x]\}$
  $c_i \leftarrow p_i \oplus k_i^N$ {Encrypt}
end for
return C

Figure 1.6: Chacha20 algorithm[17]

Internally, it uses the ChaCha20 : $\{0,1\}^{256} \times \{0,1\}^{96} \times \{0,1\}^{32} \to \{0,1\}^{512}$, $(K, N, \gamma) \mapsto k_\gamma^N$ block function in a way reminiscent of a block cipher's counter mode: the plaintext is encrypted by simply chopping P into 64-byte blocks $(p_i)_{0 \leq i < 2^{32}}$ and XOR-ing them with the 64-byte output blocks k_i^N of the ChaCha20 block function, i.e. $c_i = p_i \oplus k_i^N$. Conversely, the decryption is obtained in a simple backward manner by $P_i = C_i \oplus k_i^N$. It's important to note that the ChaCha20 block function also takes a 4-byte counter (γ), which begins at zero and is incremented with each successive encrypted block . Internally, the ChaCha20 block function operates by applying the quarter round function QR : $(\{0,1\}^{32})^4 \to (0,1^{32})^4$repeatedly on a 64 byte state S. This state is organized as a 4×4 matrix, where each entry corresponds to a 32 bit word. The value at position (i, j) in the matrix is represented as S[x], with x = 4i + j, where $0 \leqslant i, j \leqslant 3$. The initial state of S is set up as follows: the first row is populated with the constants 0x61707865, 0x3320646e, 0x79622d32, and 0x6b206574. The second and third rows are filled with the 32 byte secret key K in little endian format. The word at position 12 is assigned the block counter γ, and the remaining positions are filled with the nonce N . The quarter function (i, j, k, l)=QR(a,b,c,d) acts on the state as follows:

$$\begin{cases} e = a \boxplus b \quad ; \quad h \quad = (d \oplus e) \lll 16 \\ g = c \boxplus d \quad ; \quad f \quad = (b \oplus g) \lll 12 \\ i = e \boxplus f \quad ; \quad l \quad\ = (h \oplus i) \lll 8 \\ k = g \boxplus l \quad ; \quad j \quad\ = (f \oplus k) \lll 7 \end{cases}$$

Where $\boxplus$ denotes integer addition modulo 2^{32}, $\oplus$ denotes a XOR operation, and $\lll$ n denotes a rotation of n bits to the left. The ChaCha20 block function is composed of 10 'double rounds,' alternating

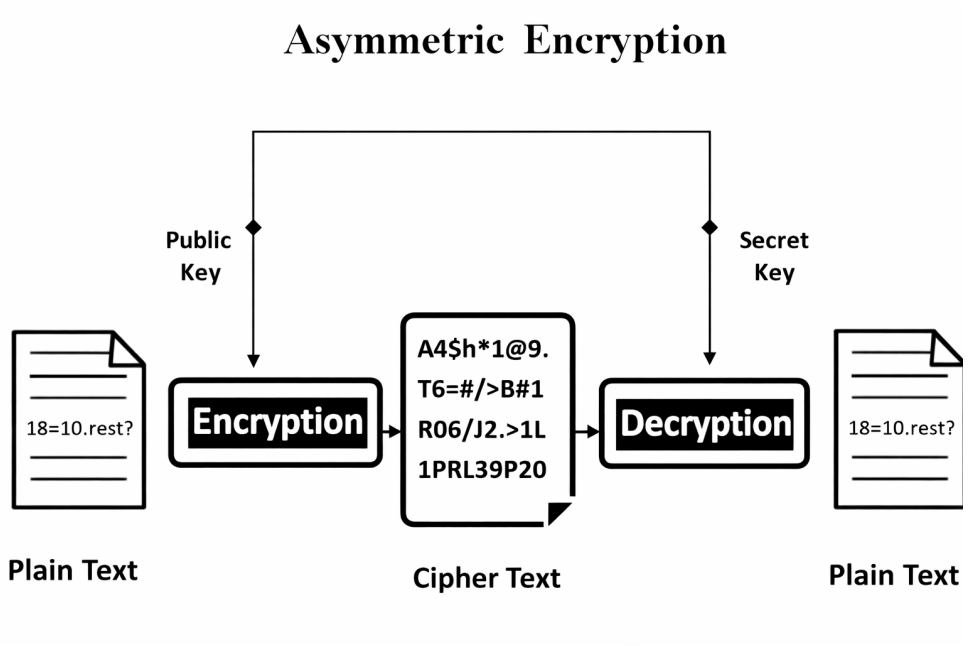

Figure 1.7: Asymmetric encryption

between a 'column round' (four quarter rounds acting on the state's columns) and a 'diagonal round' (four quarter rounds working on the state's diagonals), leading to a total of 20 rounds, or 80 quarter rounds in total. It's important to recognize that using a single key and nonce combination (along with an incrementing block counter) restricts the encryption to 256 GB of data, which is sufficient for most use cases. Additionally, it's crucial to ensure that the nonce is never reused with the same key.

5.2 Asymmetric key cryptography

Public-key cryptography, also known as asymmetric encryption, operates using two separate keys ; one for encoding and another for decoding. The key used to encode data is openly shared, while the matching private counterpart is kept secret and used to decode the information. The robustness of this approach depends on complex mathematical principles that link the key pair.

5.2.1 Elliptic Curve Cryptography (ECC)

In data security, curves based on elliptic functions are described over finite fields like $\mathbb{Z}_p$, forming sets of solutions that satisfy certain equations.[18].

$$y^2 = x^3 + ax + b \pmod{p} \tag{1.1}$$

Additionally, there is a conceptual point at infinity, labeled as $\mathcal{O}$, and the parameters a and b define

the curve within the finite field $\mathbb{Z}_p$, where

$$4a^3 + 27b^2 \not\equiv 0 \pmod{p}. \tag{1.2}$$

The elliptic curve is defined as:

$$E = \{(x, y) : y^2 \equiv x^3 + ax + b \pmod{p}\} \cup \{\mathcal{O}\}. \tag{1.3}$$

The elliptic curve supports a set of defined operations that can be performed on its coordinate points.

1. **Point Addition:** Suppose P and R are two distinct points on the elliptic curve. To compute their sum, identify where the straight line passing through both points intersects the curve again, and call this intersection Q'. Then, reflect Q' across the horizontal axis to obtain the resulting point Q, which represents $P + R$ (Figure1.8).

2. **Point Doubling:** When working with a single point P on the elliptic curve, doubling it involves computing $Q = 2P$. Instead of drawing a line through two distinct points, a tangent line is drawn at point P. This tangent intersects the curve at a second point, which is then reflected over the x-axis to obtain the result Q (Figure 1.8).

3. **Scalar Multiplication:** This refers to the process of repeatedly adding a single point P to itself on the elliptic curve. For a given integer n, the expression nP represents the result of performing the addition of P with itself n times.

The point at infinity, denoted O, functions as the identity element in elliptic curve addition; adding it to any point P on the curve leaves P unchanged.

Elliptic curve cryptographic methods operate within a cyclic subset of points on the curve, defined over a finite field. This subset is structured around a base point G, called the generator, and has a specific number of elements, known as the order n. The generator point G for the elliptic curve E is introduced in Equation

$$\begin{cases} E = \langle P \rangle \\ \forall Q \in E \ \exists n \in \mathbb{N} \text{ such that } Q = nP \end{cases}$$

The order refers to the least positive integer n for which multiplying the generator point G by n results in the identity element, known as the point at infinity (O) . The complete definition of an elliptic curve group relies on the following set of parameters, which are referred to as domain parameters [19]:

- The prime number p that indicates the order of the field Fp.
- The value a used in the curve equation.

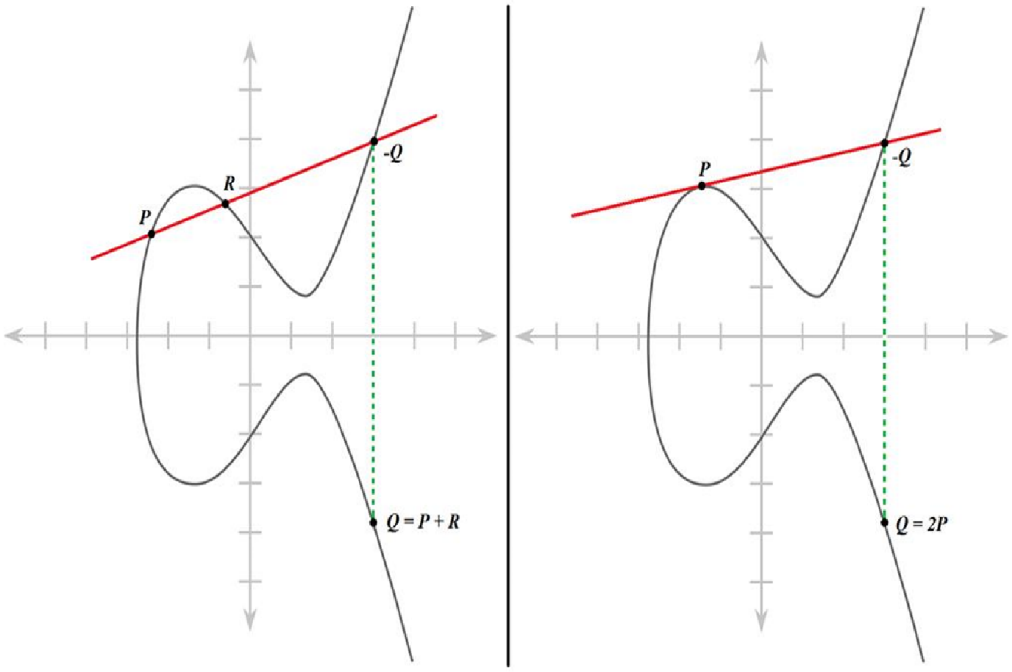

Figure 1.8: ECC Point Addition and Doubling

- The value b used in the curve equation.
- The generator G of the subgroup.
- The order n of the subgroup generated by G.

Another domain parameter, denoted as h, is used to specify the elliptic curve and represents the ratio of the total number of points on the curve to n. Hence, the EC is generally represented by the following tuple (p,a,b,G,n,h) The elliptic curves typically employed in cryptographic systems are those endorsed by the National Institute of Standards and Technology (NIST).

5.3 Lightweight cryptography

Lightweight cryptography involves adhering to specific design constraints, which various algorithms address in different ways. However, common trends can be observed, especially in the selection of non-linear operations and the design of key schedules.[20]

5.3.1 Non-linear operations

Non-linearity is essential for cryptographic strength and can be achieved through S-Boxes or non-linear arithmetic operations. S-Box-based approaches are either Look-Up Tables LUT based or bit-sliced, while ARX-based designs rely specifically on modular addition to introduce non-linearity.

- **Lut-based :**

  LUT-based algorithms utilize S-Boxes implemented via look-up tables, offering strong cryptographic properties with minimal computational effort. However, they come with drawbacks such as high memory requirements and significant vulnerability to side-channel leakage during table look-ups.

- **Bit-slice-based :**

  Bit-slice-based algorithms implement S-Boxes using parallel bitwise operations instead of look-up tables, enabling efficient and maskable software implementations. These designs require minimal logical operations and support strong security arguments, while also being well-suited for compact and efficient hardware implementations.

- **Arx-based :**

  ARX-based algorithms achieve non-linearity through modular addition and diffusion via rotation and XOR operations. While the higher-order output bits exhibit strong non-linear behavior due to carry propagation, lower-order bits remain more predictable. This creates vulnerabilities, such as differentials and linear approximations with probability 1, necessitating careful analysis of the linear components.

5.3.2 Key schedule

Lightweight algorithms significantly simplify key schedules compared to standard cryptographic designs to reduce memory and hardware costs. Unlike traditional ciphers, which can afford complex key schedules, lightweight ciphers often prioritize efficiency over resistance to related-key attacks, accepting simpler key derivation methods as a trade-off.

5.3.3 Two faces for lightweight cryptography

Defining "lightweightness" in cryptography is challenging due to the diverse requirements and implementations it encompasses. As a result, it is more appropriate to view lightweight cryptography as comprising two distinct subfields, each addressing different constraints and application needs.

- **Ultra-Lightweight Cryptography :**

  This category of lightweight algorithms targets the most constrained environments, where limited device power and connectivity are key defining factors. These constraints heavily influence the design choices made in such cryptographic solutions.

  Definition :

Ultra-lightweight cryptographic algorithms are designed for inexpensive, offline devices with limited lifespans and easy replaceability, prioritizing minimal resource usage over long-term or high-assurance security. [20]

Key features :

In ultra-lightweight contexts, strict implementation constraints justify specific trade-offs, such as accepting lower block sizes or prioritizing low-latency hardware performance. As a result, the design of such algorithms must align closely with the unique demands of their intended environments. Below are desirable features for an optimal ultra-lightweight cryptographic solution:

- Cipher mode: block or stream, offering adaptability and minimal resource use
- Minimum block length: 64 bits, ideally more
- Key length: no less than 80 bits, preferably higher when feasible
- Relevant attacks: In ultra-lightweight cryptographic settings, attackers are assumed to have limited access to data due to the constrained nature of the devices. This justifies focusing on low-data-complexity attacks, allowing for a reduced number of rounds in the algorithm while still maintaining adequate security, ultimately enhancing performance on low-power hardware.
- sca resilience : The design should facilitate the straightforward integration of countermeasures against side-channel attacks.
- Use of non-volatile memory: Storing the key in non-volatile memory is more cost effective, although it may necessitate additional attention during the algorithm's design.
- Functionality: Each device performs only a single operation type—such as a block cipher in a smart card used solely for a challenge-response protocol—eliminating the need for a hash function or MAC. Consequently, the flexibility of the cryptographic primitive is less critical in this scenario.
- Implementation flexibility: The algorithm should be adaptable to optimize specific efficiency metrics. For instance, if low latency is required, the algorithm should support it, even if it means sacrificing some area or throughput. Similarly, achieving minimal area should also be feasible.

► **IoT Cryptography :** The second subfield of lightweight cryptography focuses on IoT cryptography, which specifically addresses the security needs of internet-connected devices.

Definition :

IoT cryptographic algorithms are designed for low-power, network-connected devices, requiring a unified suite of algorithms across all devices. Given the potential for physical attacks on some

devices, such as outdoor security cameras, it is essential that side-channel attack countermeasures be straightforward to implement. [20]

Key features :

As IoT devices rely on multi-purpose microcontrollers for cryptographic operations, software efficiency becomes crucial. Many lightweight algorithms are specifically designed with a focus on optimized software implementation rather than hardware. Below the essential characteristics that an IoT algorithm must possess :

- Type: In IoT devices, a versatile cryptographic primitive, such as a block cipher or sponge, is essential to support various operations. Since the target is a microcontroller, the key cannot be hardwired but must reside in registers, which may also contribute to forming the larger internal state of a sponge.
- Block size: A minimum of 96 bits for the internal state and key is required, with larger sizes preferred. Ideally, both the internal state and key should be able to fit within the registers of a typical microcontroller.
- Key size: For IoT applications, a minimum key size of 128 bits is recommended, as different key size versions of block ciphers generally offer similar performance.
- Relevant attacks: The security model for IoT cryptography must be more robust than in ultra-lightweight scenarios, as overly limiting the data available to an adversary would be too restrictive for practical use.
- sca resilience: For side-channel attack resilience, countermeasures should be seamlessly integrated into the design for ease of implementation.
- Use of non-volatile memory: In this context, non-volatile memory plays a smaller role, as the primary focus is on optimizing software implementation.
- Functionalities: Devices that communicate complex data with various actors need multiple cryptographic functionalities, such as encryption, authentication, and hashing, to ensure secure interactions.
- Flexibility: The algorithm must be efficient across a broad range of microcontrollers, with hardware efficiency being secondary but beneficial for potential hardware acceleration.

Below, we will present some lightweight algorithms :

5.3.4 DESL and DESXL

DESL is a lightweight adaptation of the traditional DES algorithm, while DESXL is a streamlined version of the DESX algorithm. Both DESL and DESXL utilize a single S-box (substitution block) in contrast

to the original DES, which employs eight S-boxes. This design choice not only conserves memory but also enhances resistance against common cryptanalytic attacks, thanks to the S-box's robustness. [21]

5.3.5 Curupira

The Curupira algorithm is built upon Joan Daemen's Wide Trail strategy, designed to meet lightweight criteria. It boasts the following characteristics: a data block size of 96 bits, represented as a 3 X 4-byte array, with key lengths available in 96, 144, or 192 bits. The number of rounds is adjusted according to the key length. Additionally, the algorithm implements an 8 X 8-bit S-box as two 4 X 4-bit S-boxes, effectively reducing the storage space needed for the S-boxes.

5.3.6 Katan and Ktantan

KATAN and KTANTAN belong to a family of hardware-oriented block ciphers, comprising three variants each: KATAN32, KATAN48, and KATAN64, as well as KTANTAN32, KTANTAN48, and KTANTAN64. The number in the algorithm's name denotes the block size in bits, with both utilizing an 80-bit key size. However, KTANTAN differs in hardware compactness, featuring a key burnt into the target device, making it immutable. Consequently, KTANTAN is ideal for devices initialized with a single key and is more compact compared to KATAN. These algorithms possess low resource requirements due to several factors: their internal state size matches the block size, they process small data blocks ranging from 32 to 64 bits, and KTANTAN's key schedule is simplistic. Additionally, they leverage shift registers and feedback functions, facilitating straightforward hardware implementation and providing necessary nonlinearity. [22]

5.3.7 Present

PRESENT stands out as one of the most efficient lightweight algorithms and has earned ISO/IEC standard recognition for lightweight cryptography. Derived from the transformation layers of Serpent and DES, PRESENT has undergone thorough analysis, particularly regarding security and hardware efficiency. Notably, it exhibits the following characteristics that contribute to its lean profile: minimal gate count and memory usage, 31 rounds on 64-bit data blocks, support for 80 or 128-bit keys, and a remarkably compact hardware implementation requiring 1570 (GE). This places PRESENT competitively alongside today's leading compact stream ciphers, which typically demand 1300-2600 GE. While initially designed for hardware performance, PRESENT can also be implemented in software. Its primary applications involve encrypting small or moderate amounts of data. [23]

5.3.8 Hummingbird

Hummingbird is a hybrid algorithm that combines elements of both block and stream ciphers. It operates on 16-bit data blocks, employs a 256-bit key, features an 80-bit internal state, and relies on straightforward logic and arithmetic operations. Due to its small block size, Hummingbird meets minimal response time and power consumption requirements, making it suitable for RFID tags or wireless sensors without requiring modifications to current standards. However, compared to PRESENT, Hummingbird exhibits higher latency and execution times, resulting in slower encryption speeds and reduced efficiency for authentication mechanisms. Hummingbird-2, an enhanced version, introduces optional message authentication capabilities. It operates on 16-bit blocks, utilizes a 128-bit key, and initializes its 128-bit internal state with a 64-bit initialization vector. To authenticate associated data transmitted with ciphertext, Hummingbird-2 [24] employs Authenticated Encryption with Associated Data, processing associated data only after encrypting the entire payload. Notably, Hummingbird-2 excels in low power consumption and faster processing speeds.

5.3.9 Simon and Speck

Simon and Speck comprise a family of lightweight block ciphers developed by the National Security Agency (NSA) and introduced in June 2013. These algorithms are designed to serve as versatile block ciphers suitable for various IoT applications in the future. While Simon is optimized for hardware implementations and Speck is tailored for software implementations, both offer notable advantages. They exhibit excellent performance on both hardware and software platforms, boast simple construction facilitating efficient implementations, provide flexibility to construct various implementations on a given platform, and are open to analysis using existing techniques. Each family includes ten distinct block ciphers featuring differing block and key sizes. Simon variants are denoted as Simon2n, where n represents the number of bits in the block (2n-bit), while Speck follows a similar naming convention. The block and key sizes span from small to large, ranging from a 32-bit block with a 64-bit key to a 128-bit block with a 256-bit key. [25]

5.3.10 LED

Light Encryption Device (LED) is a symmetric block cipher specifically designed to be lightweight and efficiently implementable in hardware. One notable application of LED is in securely storing and transmitting data from RFID tags. LED operates with a block size of 64 bits and supports key lengths of either 64 bits (LED-64) or 128 bits (LED-128). Additionally, LED allows for key lengths between 64 and 128 bits, with any additional bits beyond the chosen length being padded with the prefix of the key. LED can also be implemented in software environments.[26]

5.3.11 TEA

The Tiny Encryption Algorithm (TEA) was specifically developed to cater to low-performing small computers, aiming for efficiency while maintaining simplicity. This block cipher operates on 64-bit blocks, which are further divided into 32-bit blocks, and utilizes a 128-bit key length. TEA follows a round-based encryption approach, with the number of rounds being variable, although 32 Tea cycles are typically recommended. Despite its 32 rounds, TEA outperforms DES with 16 rounds, and it supports all modes of DES while being implementable in various programming languages. Additionally, the eXtended TEA (XTEA) algorithm builds upon TEA's foundation, employing 64-bit blocks and a 128-bit key length, with 64 encryption rounds. XTEA introduces a more complex key management system and modifies the shift, XOR, and addition operations for enhanced security. Another variant, Block TEA, eliminates the requirement for a fixed block size and can operate with blocks of any size. Notably, Block TEA doesn't necessitate an operation mode for ensuring confidentiality and authenticity, making it applicable directly to the entire message.

5.3.12 SEA

The Scalable Encryption Algorithm (SEA) is characterized by several key features, including low memory usage, a small code size, and a limited instruction set. Its design allows for flexibility to operate on various platforms, as it can be parameterized according to the processor size, plaintext size, and key size. SEA, built upon the Feistel structure, stands out as the most compact cipher primarily due to its utilization of a 3-bit S-box. It is particularly recommended for small encryption routines where efficiency and compactness are crucial considerations. [27]

5.3.13 TWINE

TWINE, introduced by Tomoyasu, utilizes a Generalized Feistel Structure (GFS) that allows for compact implementations in both hardware and software environments. It can be efficiently implemented in hardware with only 1.5 KGates and is compatible with low-end microcontrollers, thanks to its minimal memory requirements. However, to ensure sufficient security, TWINE necessitates multiple iterations. To address this, TWINE incorporates an enhanced variant of GFS, optimizing it to achieve ultra-lightweight characteristics while maintaining adequate speed. TWINE adopts a Type-2 generalized Feistel structure and boasts a 64-bit block size with 36 rounds of encryption. It is available in two variants: TWINE-80 and TWINE-128, offering key sizes of 80 bits and 128 bits, respectively. [28]

5.3.14 Other algorithms

Several noteworthy lightweight encryption algorithms exist in the field of cryptography. One such algorithm is Skipjack[29] , a block cipher developed by the U.S. National Security Agency (NSA). Utilizing an

unbalanced Feistel network, Skipjack operates on a 64-bit block length and employs an 80-bit key. Another prominent algorithm is NOEKEON [30], known for its hardware efficiency and devised by Daemen et al. Additionally, HIGHT[31], designed by Hong et al., offers a generalized Feistel-like structure with a 64-bit block length and a 128-bit key, making it suitable for low-cost and low-power implementations. Another notable mention is mCrypton [32], a variant of Crypton tailored for compact implementations in both hardware and software. Lastly, KeeLoq, proposed by Bogdanov, features a 32-bit block size and a 64-bit key, finding widespread use in remote keyless entry systems and wireless authentication applications despite its relatively short key size. These algorithms contribute to the diverse landscape of lightweight encryption solutions catering to various security and efficiency requirements.

6 Conclusion

In this chapter, we defined the Internet of Things (IoT), the main technologies used in this context, and the various application domains of IoT. Additionally, we highlighted the different challenges and issues that need to be addressed, which are being taken seriously by the scientific community, particularly security concerns.

In the following sections, we will focus on security aspects in IoT, specifically identity management of an object and authentication of identities in this context, which form the core of our research problem.

Authentication and identity management in the internet of things

1 Introduction

As the Internet of Things (IoT) continues to expand, connecting billions of devices across various domains, the issues of identity and authentication become increasingly vital. Unlike traditional computing environments, IoT systems are highly dynamic, decentralized, and resource-constrained, which presents unique challenges for managing identities and securing communications. Ensuring trust, privacy, and secure access in such a diverse ecosystem requires robust identity management and authentication mechanisms.

In this chapter, we will define the concept of identity in the Internet of Things (IoT), as well as the requirements of identity management systems. We will also study the different classic identity management systems on the Internet and blockchain technology. Then, we will present a state-of-the-art overview of authentication protocols proposed in the context of IoT, followed by a classification and critical analysis of the proposed schemes.

2 The principle of identity in logical philosophy

The concept of identity has its roots in classical logic, beginning with Aristotle's Law of Identity and evolving through Leibniz's formulation of the Identity of Indiscernibles. Together, these principles establish identity as an equivalence relation reflexive, symmetric, and transitive and provide foundational criteria for distinguishing individuals both in the physical world and in cyberspace, including contexts like the Internet of Things [33].

Principle 1. For any x and y, if x is identical to y, then x and y have all the same properties.

$$\forall x \forall y \left(x = y \Rightarrow \forall P \left(P(x) \Leftrightarrow P(y)\right)\right) \tag{2.1}$$

Principle 2. For any x and y, if x and y have all the same properties, then x is identical to y.

$$\forall x \forall y \left(\forall P(P(x) \leftrightarrow P(y)) \Rightarrow x = y\right) \tag{2.2}$$

Although classical identity principles like Leibniz's Law offer a logical foundation, they fall short in complex, dynamic environments such as the Internet of Things (IoT). The Ship of Theseus paradox illustrates that identity cannot be fully defined by static attributes alone. In practice, current identity management systems still rely heavily on attribute-based identification and authentication methods, leading to fragmented identities, overcollection of personal data, and increased risks of identity theft. This highlights the urgent need for more robust, context aware, and privacy-preserving identity models in the interconnected digital era.

2.1 The requirements of identity management systems

With the advent of IoT, existing identity management systems on the Internet cannot be directly transplanted to IoT environments due to some native IoT characteristics such as scalability, interoperability, and mobility. These requirements are of great importance with an impact on the design of identity management systems for the IoT: [33]

2.1.1 Scalability

The scalability of identity management in the Internet of Things (IoT) era necessitates a departure from traditional centralized approaches. With billions of interconnected entities, relying on a single universal third party becomes impractical. Instead, federated identity management solutions like SAML and Shibboleth offer promise by facilitating interoperability between different identity management systems. However, in trustless networks, establishing mutual trust relationships between diverse IdM systems is essential. Thus, scalable and trusted identity management solutions are imperative for distributed trustless networks, where centralized control is absent.

2.1.2 Interoperability

The heterogeneity of IoT objects, stemming from their diverse communication technologies, protocols, and hardware, presents significant interoperability challenges. Attempts to unify identities across different manufacturers, vendors, and standard groups have been hindered by fragmented standards and divergent vocabularies. Despite initiatives like OneM2M and IoT reference architectures, the lack of an integrative approach hampers progress in addressing software, hardware, and communication heterogeneity. Therefore, achieving interoperable identity management for IoT objects requires a concerted effort towards standardization and integration across the spectrum of technologies and stakeholders involved.

2.1.3 Mobility

The pervasive mobility of IoT devices, including vehicles and wearables, underscores the necessity for identity management systems that can accommodate continuous connectivity and access control, regardless of location. Users expect seamless authentication, authorization, and access control for device services, irrespective of their mobility. Thus, effective identity management for the IoT must prioritize mobility and offer peer-to-peer authentication and authorization services to ensure seamless and secure connectivity in dynamic environments.

3 Classic identity management systems on the internet

Traditional Identity Management (IdM) systems operate through a trust-based model involving three interdependent entities[34]: the subject (user), the relying party (service provider), and the identity provider (IdP). The identity provider plays a central role by authenticating the user and asserting their identity to the service provider, enabling secure access based on the user's verified attributes such as roles, positions, and privileges.

Over the past thirty years, Identity Management Systems (IdMs) have progressed from isolated to centralized and subsequently to federated models. [35]

The isolated Identity Management (IdM) model entails each application or system managing its user identities independently without centralized coordination. In this model, each application typically operates with its user database and authentication mechanism. Key characteristics of this model include decentralization, leading to each system managing identities and access control policies independently, resulting in duplication of effort and potential inconsistencies. Integration between systems is limited, posing challenges in maintaining a unified view of users across the organization. Additionally, security risks arise due to inconsistent access control policies and varying security measures. As the organization expands, scalability becomes a challenge. In contrast, centralized IdM models consolidate user identities and access control mechanisms into a centralized system, offering a unified and standardized approach to identity management across the organization.

The centralized Identity Management (IdM) model in IdM systems involves managing user identities and access control policies from a single, centralized location within an organization. This model is characterized by a singular authority responsible for authenticating users, managing their identities, and enforcing access control policies across various applications and systems. Key features include a centralized authority for managing user identities and access control, a unified repository storing user identities and credentials, Single Sign-On (SSO) capabilities for seamless authentication across multiple systems, consistent enforcement of access control policies, scalability and efficiency in managing large user bases, and enhanced security measures such as strong authentication mechanisms and centralized

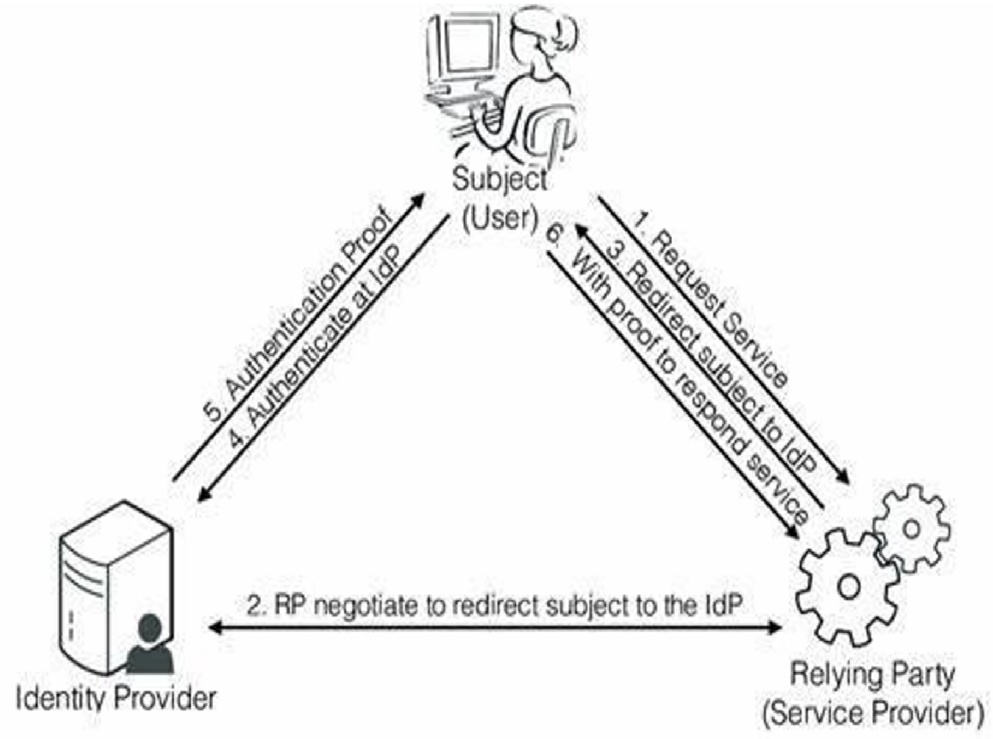

Figure 2.1: StakeholdersfromthetraditionalIdMSmodel[33]

audit trails. Overall, the centralized IdM model offers organizations improved control, efficiency, and security in managing user identities and resource access across their IT infrastructure.

The federated Identity Management (IdM) model in IdM systems involves the sharing of user identities and access rights across multiple trusted domains or organizations. Unlike the centralized model, where control is singular, federated IdM allows for decentralized identity management while ensuring seamless and secure resource access across organizational boundaries. Key features of this model include interoperability, where identity information flows seamlessly across domains; trust relationships established through agreements or protocols like SAML or OpenID Connect; support for Single Sign-On (SSO), enabling users to authenticate once and access resources across federated domains without reauthentication; distributed identity sources managed locally within each organization but seamlessly integrated through trust relationships; compliance with privacy regulations and policies; and flexibility and scalability, empowering organizations to collaborate and share resources while retaining autonomy over identity management processes. Overall, the federated IdM model facilitates effective and secure collaboration by enabling users to access resources across diverse domains or organizations using their existing identities and credentials.

From an industrial standpoint, numerous initiatives and groups are actively engaged in identity management, as outlined in Table 1, encompassing entities like PRIMELife[36], SWIFT[37], DAIDALOS[38], Kantara[39] (formerly Liberty[40]), FIDIS[41], SAML[42], Higgins[43], OpenID[44][44], Shibboleth[45],

STORK[46], PICOS[47], and Cardspace[48], among others. These initiatives are assessed against IoT identity management requirements such as scalability, interoperability, mobility, security, and privacy.

The comparison in Table 2-1 reveals that many of these systems struggle with scalability, as users are tasked with integrating various entities from IoT and coordinating across different application domains to engage with the Identity Management Systems (IdMS). This compromises scalability and complicates the creation of an interoperable system in such a heterogeneous environment. While some initiatives like OpenID or PICOS exhibit extensibility due to their decentralized architecture, a robust extensible system is still necessary for managing all entities in the IoT environment. Additionally, mobile IdMS is essential in the ubiquitous IoT landscape, ensuring user identity accessibility regardless of location. While most initiatives prioritize security and privacy, they often rely on users trusting their Identity Providers (IdPs), undermining user privacy. Despite adopting a user-centric model, these systems still depend on third-party identity providers, perpetuating privacy concerns.

In summary, contemporary identity management systems on the Internet have progressed from isolated, attribute-based models to decentralized, federated, and user-centric solutions like OpenID, embraced by major online service providers such as Facebook and Google. Typically, the IdMS of prominent online service providers becomes the universal identity provider within their federated domains. While federated user-centric IdMS simplifies identity management for users, security and privacy concerns persist, as users must place full trust in their identity providers, who can observe all user activities with online service providers. Hence, eliminating unnecessary third parties and establishing a trusted identity provider in trustless networks are crucial for the IoT environment.

	Scalability	**Interoperability**	**Mobility**	**Security & Privacy**	**User-Centric**
PRIMELife (PRIME)			*	*	*
SWIFT (DAIDALOS)		*	*	*	*
Kantara (Liberty)	*	*	*	*	
FIDIS		*	*	*	*
SAML		*			
Higgins		*		*	*
OpenID	*				*
Shibboleth				*	
STORK		*		*	*
PICOS	*	*	*	*	*
Cardspace		*	*	*	*

Table 2.1: Comparison of Identity Management Systems

Current identity management systems, while evolving toward decentralized and user-centric models like OpenID and federated solutions, still face significant challenges in the Internet of Things (IoT) environment. These systems struggle with scalability, interoperability, and extensibility, particularly due to the complexity and heterogeneity of IoT entities. Moreover, privacy concerns remain unresolved because

users must still trust third-party identity providers that are involved in every transaction, potentially compromising user data.

To address the demands of ubiquitous and mobile IoT environments, there is a critical need for a more robust, extensible, and privacy-preserving identity management system one that minimizes reliance on centralized or third-party providers and supports secure identity access in trustless networks.

4 Blockchain technology

Blockchain technology, originally introduced along side the Bitcoin cryptocurrency, presents diverse opportunities and research avenues in the realm of IoT security. Serving as distributed ledgers, blockchains maintain immutable records of all transactions within the Bitcoin peer-to peer network, defining transaction organization, block creation, confirmation, and storage protocols. Mining, employing the Proof of Work (PoW) mechanism introduced by Nakamoto, ensures transaction consistency and addresses the double spending issue in decentralized networks, eliminating the need for a centralized authority like a bank. Each node in the network maintains a tamper-proof copy of the blockchain, which is updated simultaneously across the network for instant transaction validation. With Ethereum leading the blockchain's evolution into its 2.0 era since 2014, a decentralized platform enables the creation of decentralized applications (Dapps) via smart contracts, ensuring resilience against downtime, censorship, or fraud. Ethereum's Turing-completeness smart contract concept fosters the development and deployment of various Dapps. Blockchain's ability to decentralize networks by removing intermediaries opens avenues for numerous initiatives and research in IoT security, prompting a survey of its potential to revolutionize identity management systems. [33]

4.1 Blockchain- an overview

Blockchain has emerged as a key innovation in distributed systems, enabling secure and transparent data management across peer-to-peer networks without relying on centralized entities . In October 2008[49], blockchain technology was first presented as a foundational component of Bitcoin, a digital currency system designed to operate without relying on a central governing body for tasks such as currency issuance, ownership transfer, and transaction verification.

The blockchain ledger is made up of a series of blocks. Each block consists of two main components:

- The body of the block, which holds the transactions (referred to as 'facts') that need to be stored in the database. These facts can vary, including financial transactions, medical records, industrial data, system logs, and more.
- The block's header, which includes metadata about the block, such as the timestamp, transaction hash, and other details, along with the hash of the preceding block.

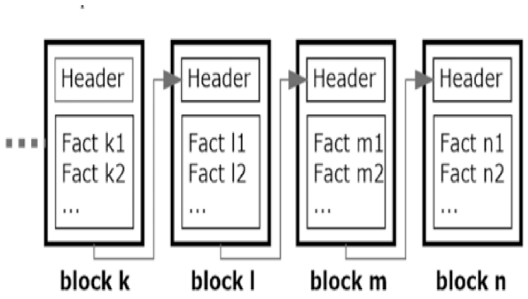

Figure 2.2: blockchain example

Consequently, all the blocks are connected together, creating a sequence of linked and organized blocks. The more extended the chain, the more difficult it becomes to alter. If an attacker attempts to change or replace a transaction in a block, (1) they would need to modify every subsequent block, as each is linked through its hash. (2) Additionally, they would have to update the version of the blockchain stored by each participant. Figure (2.2) illustrates a simplified example of a blockchain.

4.2 Architecture

There are different types of nodes in a blockchain [50]: Full Nodes, Lightweight Nodes, and Mining Nodes.

1. **Full Nodes:** Full nodes are tasked with storing and distributing copies of the entire blockchain. They download all blocks and transactions, verifying them according to consensus rules. A full node can validate transactions all the way from the first block, known as the genesis block.

2. **Lightweight Nodes:** Also referred to as light nodes, these nodes don't store the entire blockchain. Instead, they only download the block headers, which they use to verify the validity of transactions.

3. **Mining Nodes:** Commonly called "Miners," these nodes are responsible for creating new blocks for the blockchain. They validate and confirm blocks to be added through the process known as mining.

4.3 Key characteristics

Blockchain has the following key characteristics [51] :

- **Decentralization:** In traditional systems where transactions rely on a central authority (such as a bank), each transaction requires validation by this entity, often leading to increased costs and performance delays at the central point of control. On the other hand, blockchain allows transactions to take place directly between users, eliminating the need for central validation. This

decentralized model helps reduce the overhead costs of server infrastructure, including both initial setup and ongoing maintenance, while also removing the performance limitations associated with central servers.

- **Persistence:** Because every transaction must be verified and stored in blocks that are distributed throughout the entire network, altering them is virtually impossible. Furthermore, each broadcasted block undergoes validation by other nodes, ensuring that transactions are thoroughly checked. As a result, any attempt at falsification can be quickly identified.
- **Anonymity:** Users can engage with the blockchain network using unique addresses, and they have the option to create multiple addresses to protect their identity. Unlike traditional systems, no central authority holds personal data about users. This approach ensures a degree of privacy for transactions on the blockchain. However, it's important to note that blockchain cannot fully guarantee complete privacy due to its inherent limitations.
- **Auditability:** Every transaction on the blockchain is verified and timestamped, allowing users to effortlessly check and trace past transactions by accessing any node in the decentralized network. In the Bitcoin blockchain, each transaction can be linked back to prior ones, creating a clear chain of events. This enhances both the traceability and transparency of the data stored on the blockchain.

4.4 How Blockchain works?

Process of Adding a Block: To add a new block to the blockchain, one have to follow the following steps:

1. A set of transactions is collected and bundled together in a block.
2. Miners confirm that the transactions in the block adhere to the specified rules.
3. Miners then apply a consensus protocol to authenticate the new block.
4. Those who successfully validate the block are rewarded.
5. Once validated, the transactions are permanently recorded in the blockchain

Network Operation Process [52]: The steps to run the network are as follow

1. New transactions are sent out to all nodes in the network.
2. Each node gathers these transactions and forms them into a block.
3. Following the blockchain's consensus protocol, nodes collaborate to verify and integrate the new block.

4. A block is only accepted by nodes if all its transactions are legitimate and not previously spent.

5. Once a block is accepted, nodes indicate their approval by starting work on the next block, referencing the accepted block's hash as the previous one.

4.5 Consensus algorithms

Consensus Protocols in Blockchain: Blockchain systems rely on various consensus mechanisms to ensure the legitimate validation of blocks. The effectiveness and suitability of a consensus protocol are determined by three key attributes [53]:

1. **Safety:** A protocol is considered safe if all nodes reach the same valid result, where the validity of the outcomes follows the established rules of the system. This is also known as maintaining consistency in the shared state.

2. **Liveness:** Liveness is achieved when all non-faulty nodes involved in the consensus process eventually produce a valid output.

3. **Fault Tolerance:** A protocol is fault-tolerant if it can continue functioning properly even after the failure of any node involved in the consensus process.

There are various methods for validating transactions, with Proof-of-Work (PoW) and Proof of-Stake (PoS) being the most commonly used.

Proof-of-Work is the consensus mechanism implemented in the Bitcoin network [49]. In PoW, every node in the network performs calculations to find a hash value for the block header, which changes as new transactions are added. The protocol requires that this hash be less than or equal to a predetermined target. Miners must repeatedly calculate different hash values using various nonces until they reach the target. Once a miner succeeds, other nodes in the network verify the correctness of the computed value. This process ensures the validity of the transactions in the block, especially in the case of fraudulent activity. After verification, the set of transactions is accepted as valid, and a new block is added to the blockchain. Due to the computational power and time required for this process, an incentive, such as a reward in cryptocurrency (e.g., Bitcoin), is typically offered to miners [49][51][53].

Proof-of-Stake, on the other hand, is a consensus approach used to achieve distributed consensus in cryptocurrency networks. In PoS, a node is randomly chosen to add a new block to the blockchain based on the amount of cryptocurrency it holds. The node must prove it owns the cryptocurrency linked to the blockchain to participate in this process. The likelihood of being selected is proportional to the quantity of cryptocurrency the node possesses. One of the primary benefits of Proof-of-Stake is that it eliminates the high energy costs associated with Proof-of-Work. However, critics argue that PoS may not offer the same level of security or immutability as the Proof-of-Work method used in Bitcoin. As a result, some research is exploring hybrid approaches that combine PoW and PoS mechanisms.

Various consensus algorithms offer distinct benefits and challenges. Delegated Proof of Stake (DPoS), introduced by Daniel Larimer in 2014 [54], shares similarities with Proof of Stake (PoS) but introduces a system where miners are selected to create blocks based on the amount of stake they hold.

Other consensus mechanisms are based on variants of Byzantine Fault Tolerance (BFT). One notable example is Practical Byzantine Fault Tolerance (PBFT) , which was the first effective solution for achieving consensus in environments prone to Byzantine failures. PBFT is used in platforms like Hyperledger Fabric. Another variation, Cross Fault Tolerance (XFT) , simplifies the attack model, making Byzantine Fault Tolerance both practical and efficient for real-world applications. Two blockchain-based platforms, Ripple and Stellar, utilize modified versions of BFT. These platforms enable open-ended node participation by employing their respective consensus algorithms: the Ripple Consensus Protocol and the Stellar Consensus Protocol.

4.6 Blockchain platforms

The number of blockchain platforms is constantly increasing, with some of the most prominent being Bitcoin, Ethereum, and Hyperledger Fabric.

Bitcoin [49] is a decentralized system designed for transferring and verifying digital currency. It operates on a peer-to-peer (P2P) network without the need for any central authority. Bitcoin transactions are timestamped and secured through a continuous chain of hashes using proof-of work. Initially, the protocol was created as an experimental electronic payment method relying on cryptographic proof rather than trust. The smallest unit of Bitcoin is denoted in eight decimal places, and the currency itself is called bitcoin. Bitcoin software allows users to generate payment addresses for sending and receiving bitcoins, and the source code is open and accessible to the public.

Ethereum is considered the most promising blockchain platform after Bitcoin. Its creators describe it as the ”first true global computer” [55]. Ethereum supports a cryptocurrency known as Ether (ETH) and is designed not only for financial transactions but also for processing various other types of data. Beyond block validation, Ethereum miners also work with Smart Contracts, which are programmable sets of rules that run directly on the blockchain [56]. This enables Ethereum to serve as a platform for decentralized applications. Smart contracts are executed using the Ethereum Virtual Machine (EVM), which is the underlying operating system.

Hyperledger Fabric [57] is a distributed ledger platform that uses a modular architecture, ensuring high levels of confidentiality, resilience, flexibility, and scalability. It is designed to support customizable implementations of various components, making it suitable for complex use cases across different industries. Hyperledger Fabric stands out due to its adaptable and extensible design, distinguishing it from other blockchain solutions. Its open-source framework is fully vetted, ensuring reliability for enterprise-level blockchain applications. The platform uses a Byzantine Fault Tolerance (BFT) consen-

sus mechanism for validating transactions through a replicated state machine.

5 Blockchain based IdMS

The concept of blockchain-based identity, also known as self-sovereign identity, shifts control of access rights and identity management from traditional providers to individuals themselves, preventing attacks from malicious third parties. This approach can be implemented through permissioned (e.g., Sovrin) or permissionless (e.g., Uport) methodologies. Essentially, individuals and collectives can selectively store identities in the blockchain without compromising privacy and issue claims to each other. These claims, which act as endorsements, can come from various entities such as governments, banks, or universities. Individuals and collectives gradually complete their identities by adding attributes and submitting related information to create blockchain identities. This process involves generating identity attributes, creating blockchain identities, and issuing claims using public and private key pairs. For instance, an individual like Alice can self-generate attributes and receive claims from various sources like her employer, bank, or government. In different situations, Alice can present these claims to verify her identity or qualifications, such as when applying for a job or receiving salary payments.[33]

Table 2.2: Blockchain identity management solutions

	DNS	PKI	Storage	Bitcoin Based	Ethereum Based	Full Stack	Reputation	Privacy	Year
Namecoin		*		*					2014
Certcoin		*	*						2014
Fromknecht		*		*					2014
Uport									2015
Sovrin		*						*	2016
Jolocom		*			*				2016
Blockstack		*	*	*					2016
Authcoin		*		*					2016
ChainAnchor		*					*	*	2016
Liu et al		*					*		2017
NEXTLEAP		*							2017
Azouvi		*			*				2017
Axon		*		*					2017
Augot		*		*					2017
SCPKI		*					*		2017

6 A sliver of light

Self-sovereign blockchain-based identity management systems eliminate centralized identity providers by creating identities on the blockchain platform, allowing users and service providers to follow identity consensus and verify identities independently. The concept of claims in blockchain Identity Management Systems (IdMS) extends from relationships in federated identity management, serving as endorsements

crucial in trustless distributed blockchain-based IdMS. Digital identities facilitate communication and access to applications and services, reflecting the complexity of social relations in human society. Similarly, the Social Internet of Things (SIoT) applies social network principles to IoT, transitioning from isolated devices to unified devices forming social-like networks. While SIoT focuses on device social networks, security architecture for IoT integrates blockchain-based social networks, unifying IoT entities and addressing security issues through three-dimensional social networks. Root identities represent individuals or collectives, while partial identities denote IoT services provided by devices. Relationships between subjects and things, managed by blockchains in trustless peer-to-peer networks, enable configuration of access control security policies. The socialized IoT paradigm resolves security and trust issues in a distributed trustless IoT environment, while addressing scalability, heterogeneity, and mobility challenges through the inherent characteristics of social networks.[33]

7 Authentication in internet of things

Among the security issues related to IoT, authentication is a very important security aspect that must be implemented effectively. Authentication allows the verification of each identity of a connected object . However, the design and implementation of an authentication scheme must be adapted to the limitations and constraints of IoT. An authentication scheme primarily aims to prevent any unauthorized data transmission within a system by a malicious object, as any alteration or modification of transmitted data could lead to a malfunction.

Since the emergence of IoT, several research studies on the authentication of object identities have been conducted. Their objective is to propose new lightweight authentication and key establishment schemes adapted to IoT, so that an authentication scheme is reliable, protected against various possible attacks, efficient in terms of computation and execution time. An authentication scheme deployed in one or more IoT application domains must also meet the specific needs of the use case. As a result, the reliability and effectiveness of a security protocol in this field are essential [1][2]

We have studied several works and research projects on identity authentication in an IoT context, such as the authentication schemes proposed for Identity-Based Encryption, Smart Home, and ECC-based authentication. Based on our study of the state of the art, we were able to classify authentication schemes into two main categories: authentication with asymmetric encryption and authentication with symmetric encryption.

In the research conducted by Salami et al. [58], a lightweight encryption approach tailored for smart home environments is introduced, utilizing Stateful Identity-Based Encryption (IBE). This method eliminates the need for digital certificates by relying solely on identity strings as public keys. By leveraging the sender's and receiver's identities for encryption and decryption, the scheme ensures secure com-

munication between connected devices. Furthermore, its certificate-free nature enhances efficiency and simplifies implementation, making it well-suited for smart home networks and other IoT applications. Therefore, the reliability and efficiency of a security protocol in this field is essential.

The Phong, Matsuka, and Ogata (PMO) Stateful Identity-Based Encryption (IBE) model integrates IBE with the Stateful Diffie-Hellman (DH) encryption technique. It structures the encryption process into two distinct phases: key encryption and data encryption. This design enables multiple transmissions of encrypted data without repeatedly attaching the encrypted key. Performance evaluations indicate that this approach effectively resists plaintext attacks, making it a reliable solution for securing communications within IoT ecosystems.

Santoso and Vun [59] introduced a robust authentication protocol designed for IoT-enabled smart homes, utilizing Elliptic Curve Cryptography (ECC). Their proposed system incorporated a gateway-focused AllJoyn framework, enhancing the authentication process specifically for Android devices. This approach ensures secure mutual authentication between communicating entities while minimizing message overhead, improving efficiency in smart home networks.

Ola et al. [60] proposed an identity-based authentication mechanism for IoT within a Software-Defined Network (SDN) environment. This approach utilizes a virtual IPv6-based shared identity managed by the SDN controller to standardize identity translation across different communication protocols. In this framework, gateways handle device authentication, while the central controller verifies the gateways themselves. Designed to withstand prevalent security threats such as replay attacks, man-in-the-middle interceptions, and identity spoofing, this scheme enhances the security and reliability of IoT communications.

Ye et al. [61] introduced an authentication method leveraging Elliptic Curve Cryptography (ECC) to enhance security at the perception layer of the Internet of Things (IoT). While this approach improves authentication efficiency, it does not incorporate attribute-based access control mechanisms for device interactions. Exploring this aspect further could be valuable, as establishing a secure and resilient access control framework is essential for ensuring the reliable functioning of IoT systems.

Witkovski et al. [62] introduced an authentication mechanism for IoT devices that relies on session keys and symmetric encryption. This approach is particularly advantageous in smart home environments, where a gateway facilitates seamless integration between conventional internet services and IoT networks while ensuring the security of connected electronic devices. Implementing this authentication scheme can significantly enhance the overall security framework for smart home systems.

Santis et al. [17]conducted an analysis of the ChaCha20-Poly1305 authenticated encryption with associated data (AEAD) scheme on ARM Cortex-M4 processors. Their findings revealed that this encryption method outperforms hardware-accelerated AES-128 when used in GCM, EAX, and CCM modes. Due to its superior speed and efficiency, ChaCha20-Poly1305 presents a compelling option for

applications demanding fast and lightweight encryption, particularly in IoT environments.

Reza et al. [63] proposed a security approach integrating ChaCha20 encryption, chaos-based key generation, and public key authentication. Mathematical analysis indicates that this method is well-suited for Smart Grids (SGs), providing benefits such as lower power consumption, improved processing speed, and enhanced data throughput. These features make the scheme highly efficient and lightweight. Additionally, it effectively safeguards IoT networks against replay and timing attacks, ensuring a robust and secure communication framework.

7.1 Summary of Authentication Schemes in IoT

The study of different authentication schemes related to the two classes of authentication has shown that symmetric authentication offers significant advantages in terms of computational, storage, and communication costs. As a result, it provides lower energy consumption compared to asymmetric authentication. Consequently, symmetric authentication schemes are better suited for IoT environments.

Therefore, in the context of this thesis, we focus on symmetric authentication with the aim of improving the performance of these authentication schemes.

Among the authentication schemes using symmetric encryption studied, we have Witkovski et al [62], Santis et al [17], and Reza et al [63].These authentication schemes are based on cryptographic operations as well as the use of hash functions in the various authentication exchanges.

These authentication schemes offer resistance to several possible attacks, such as replay attacks. On the other hand, the scheme by Witkovski et al. [62], the scheme by Santis et al. [17], and the scheme by Reza et al. [63]are vulnerable to side-channel attacks.

In order to summarize our study of research on authentication in IoT, we have defined several key criteria for evaluating the authentication schemes proposed in this context. We observe that the evaluation of an authentication scheme is based on two analyses: security and performance.

- ✓ The security analysis is conducted by evaluating a protocol based on the security features it ensures, such as mutual authentication, key sharing, synchronization independence, etc. It also includes resistance to potential attacks, such as basic network attacks like replay attacks, identity theft, and denial of side channel attacks.

- ✓ The performance analysis is a study of an authentication scheme in terms of efficiency. It includes factors such as the execution time, and storage costs.

In the second part of this thesis, based on these evaluation criteria, we compare the authentication schemes we have proposed with other presented schemes.

8 Conclusion

In this chapter, we clarified identity management in the Internet of Things (IoT) and the use of blockchain in this domain. Furthermore, we presented several research works based on authentication in an IoT context and classified them according to the evaluation criteria of an authentication scheme.

In the following section, we will present our contributions as well as the work we have carried out to address our research problem. We will conclude by evaluating and comparing our work with other authentication schemes, in order to demonstrate the effectiveness and security of our schemes.

Contribution

and proposed

solutions

Contribution and proposed solutions

1 Introduction

In this chapter we will present our proposed work. First , we will start with the first proposition entitelled " an efficient lightweight authentication and access control for iot edge devices " [64]. A new method has been developed to manage authentication and permissions specifically for Internet of Things (IoT) edge environments. This strategy supports a wide range of edge devices and allows for efficient data transfer despite bandwidth limitations by leveraging a lightweight symmetric encryption scheme ;specifically, the ChaCha20 cipher for secure session key setup. Furthermore, it strengthens identity verification and access control through the integration of blockchain frameworks and the HoBAC mechanism.

Finally, our proposed protocol entitelled « lightweight authentication for iot edge devices » [65]. This protocol alow an edge device and edge server to authenticate each auther mutually to ensure the security of the collected data . it used the hash function and the symetric encryption to tramsmit the date during the authentication process.

2 An efficient lightweight authentication and access control for iot edge devices

Our proposed approach is designed for edge networks consisting of numerous nodes, facilitating the transmission of large volumes of data within bandwidth-limited environments. It leverages lightweight symmetric cryptography, specifically the ChaCha20 algorithm, to establish session keys. Additionally, the protocol enhances identity management and access control with greater precision through the integration of HoBAC and blockchain technology. The correctness of the protocol has been rigorously validated using the Scyther tool.

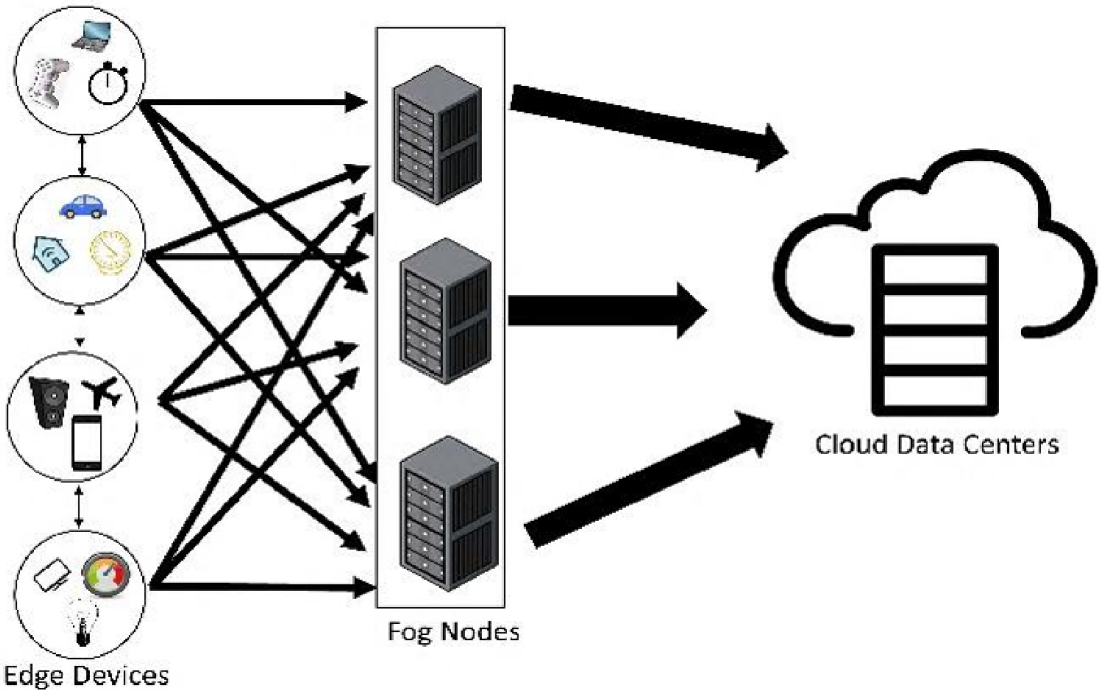

Figure 3.1: edge computing architecture

2.1 The proposed authentication scheme

We outline a framework composed of three distinct steps aimed at enhancing the security of IoT infrastructures. Initially, devices at the edge are enrolled with a central trusted authority. Next, the system confirms the legitimacy of each device through a verification step. Finally, a secure mechanism is used to generate and share encryption keys among devices. This model contributes to strengthening trust and stable communication across connected IoT elements.

2.1.1 Architecture

This study introduces a framework combining edge computing and blockchain technology to enhance the security of IoT networks. Data is gathered through an edge device functioning as a wireless sensor network (WSN), while an edge server integrates the infrastructure, supported by a series of base stations (Figure 3.1). The suggested protocol leverages blockchain technology for identity management and implements High Order Attribute-based access control to limit user permissions. Furthermore, it employs the ChaCha20 symmetric encryption algorithm to secure data. This approach could provide a robust and secure method for safeguarding data within IoT networks.

Adversary model Before presenting our proposed protocol, it is crucial to outline the adversary model utilized. The adversary, denoted as Adv, operates within polynomial time and has complete control over the vulnerable network traffic, aiming to undermine the security of the proposed system. Adv can

manipulate various aspects of message transmission over an insecure channel, including interception, modification, and deletion of messages. Additionally, Adv has the capability to extract security parameters from a smart card through power analysis methods. Adv may also employ side-channel attacks to acquire sensitive information, such as passwords. The primary objectives of Adv are as follows:

- Deriving the session key after successfully executing the authentication process.
- Determining the server's long-term secret key.
- Forcing the server to mistakenly authenticate an entity that is not legitimate.

2.1.2 Notations and abbreviations

For the reader's convenience, the notations used in our authentication scheme are defined in the following table (3.1) :

Notation	**Descriptions**
$\mathrm{ID}_{user}, \mathrm{ID}_{Node}$	The identity of user/node
$\mathrm{PK}_{user}, \mathrm{PK}_{Node}$	The public key of user/node
H_1, H_2	Hash function
Com	The commitment scheme
Γ	The group commitment
q_i	Element of the unique commitment vector
v_i	The opening
N	The number of k ($n = \|k\|$)
K	Preshared secret key
IDK	The node partial key
$\oplus$	XOR operation
$\|$	Concatenation operation
(E, D)	ChaCha20 encryption scheme
Mk	The secret key of base station
params	A set of public parameters generated by BS

Table 3.1: Notations

2.1.3 Authentication protocol

ChaCha20 offers several benefits over other cryptographic techniques while maintaining the same level of security, particularly in its suitability for devices with limited resources. As a result, a mutual authentication approach using ChaCha20 can effectively generate a secure secret key in an IoT network. This approach facilitates secure communication among devices, ensuring that data can be exchanged safely and reliably.

▶ Edge node registration phase:

H

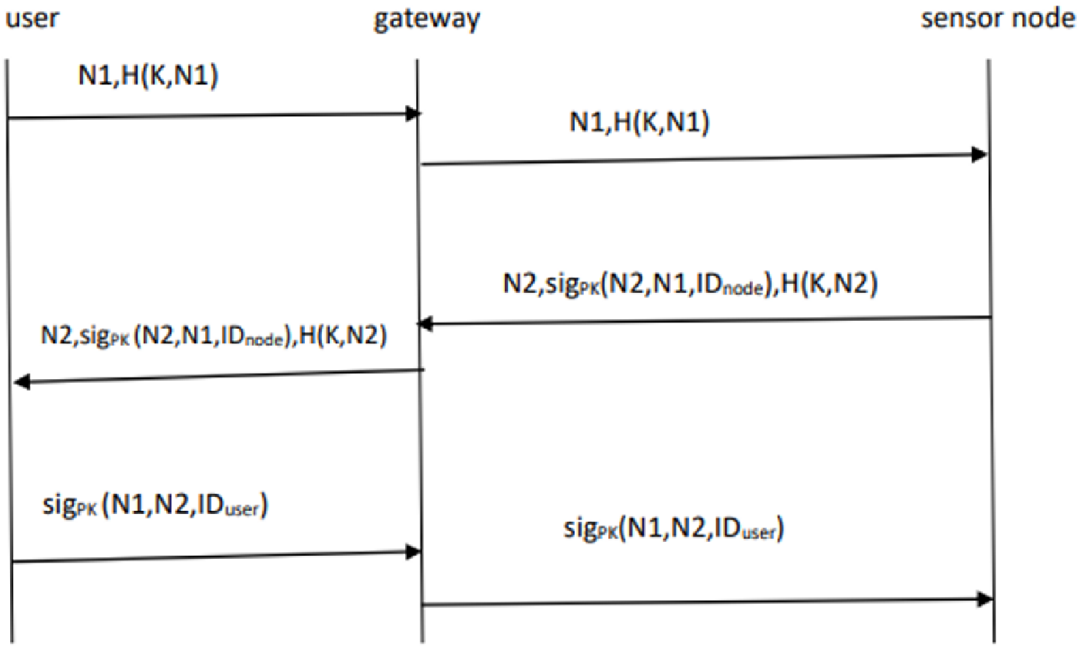

Figure 3.2: authentication phase

- Blockchain technology, in conjunction with threshold public key signatures, enables each device (whether node or user) to obtain its unique ID. This not only ensures secure data exchange between devices but also establishes a trustworthy platform for running smart contracts. Furthermore, blockchain guarantees device anonymity while fully leveraging the advantages of distributed ledger technology.
- Once the smart contract is received, the edge server stores it in a protected database to safeguard the data from malicious actions or unauthorized access. To further enhance security, the edge server implements a range of protective measures, including encryption, authentication, and authorization, to secure the stored information.

► Authentication phase :

- The user send to the node (N1, H(K,N1)).
- The node receiving the message and send (N2,sig_{pku}(N2,N1,ID_n), H(K,N2)).
- Once the user receiving the message, it send sig_{pkn}(N1,N2,ID_u). Figure (3.2)presents this process.

- K represents a pre-shared secret between two parties, playing a crucial role in securing communication between them. This secret can serve as a password or key for both authentication

and encryption, ensuring the protection of transmitted data from unauthorized access. Furthermore, the pre-shared secret can be combined with additional security measures, such as digital certificates and access control protocols, to provide enhanced protection.

- N1 and N2 are randomly generated values created by the two parties engaged in secure communication. These pseudorandom numbers serve as supplementary security elements, enhancing the overall protection of the exchange. Typically generated at the start of the communication, they are integrated into authentication and encryption procedures to further ensure the safety of the interaction.
- H refers to a hash function, a cryptographic method used to produce a distinct digital fingerprint of data. The function processes input data and generates a fixed-length output that cannot be reversed. This output is often employed for purposes like verifying data integrity and enabling secure communication. Hash functions are commonly combined with other security techniques, such as encryption and access control, to strengthen overall protection.

► Establish session key phase:

a) Setup: The base station(BS), acting as the key management servers in ATSE, generates G1 and G2, both of which have the same prime order q, and selects an admissible pairing $e : G_1 \times G_1 \rightarrow G_2$, where g is a generator of G_1. It then picks $s \in \mathbb{Z}_q$ at random and computes y = g^s . The BS also selects a symmetric encryption scheme (E,D), where E and D denote encryption and decryption procedures respectively. The BS also chooses two hash functions H1:0, 1 $* \rightarrow$ G1 and H2 :G2 $\rightarrow$ 0, 1 n for some positive integer n, which is the size of the key. The BS sets secret master key mk = s and a set of public parameters params = (G1, G2, e, g, y,(E, D), H1, H2). The BS distributes params to all of the entities involved.

b) Setup: Upon receiving an identity $id_N$the BS check whether the requester's identity is indeed idN and return IDK = e(H1(id_{Node}))^s as partiel key

c) KeyEncrypt: The sender performs the following:

- pick a symmetric key K
- ifst=null, do the following:
  - pick $r \in \mathbb{Z}_q$ uniformly at random
  - set st = (r, gr , e(H1(id_{Node}), y)^r)
- else do the following: – parse st as (r, gr , e(H1(id_{Node}), y)^r)
- pick a sample random value vi
- compute the commitment γ and q_i
- for each $i \in [n]$, sender executes:

- compute the message specific whole key wk_i = e(H1(id_{Node}), H2(com(k_i , v_i)))^s
- Generate $ck_i = (id_{\text{Node}}, H_2(\text{com}(k_i, v_i)), \gamma, q_i, e_i)$, where $e_i = \text{PRG}(wk_i) \oplus (k_i \parallel v_i)$.
- Output ck = (ck1, ck2, ..., ckN) as "key ciphertext"

d) DataEncrypt: search the symmetric key K, then perform the following:

- compute c_m=EK_m
- return cm as "data ciphertext"

e) DKeyDecrypt: upon receiving key ciphertext ck the receiver performs the following using id_{Node} :

- compute wk = (H1(id_{Node}), H2(com(k_i , v_i)))
- Decrypt e as $(k \parallel v) = \text{PRG}(wk) \oplus e$.
- Verify the commitment γ with opening v_i; if it returns 0, then output $\perp$, else output k.

f) DataDecrypt: upon receiving data ciphertext cm, the receiver finds k, performs the following using k the decrypted symmetric key

- compute m = Dk(cm)
- return m

2.2 Higher order attribute based access control model

Ensuring security in dynamic IoT environments is a prominent and challenging task. Access control is one of the essential aspects of data security, as it helps to control access to information based on pre-defined access policies. HoBAC is a new access control model that is a generalization of ABAC, and helps to create flexible access control policies for both IoT and non-IOT systems, by building hierarchies of entities based on attributes. The HoBAC model also supports fine composition and aggregation operations, allowing for highly adaptable access policies in a variety of real-world scenarios. The HoBAC model consists of three main components: entities, subjects, objects and contexts. Access control in ABAC is structured around three core component : users, resources, and environmental conditions ,which together define the actors involved in access decisions. The initiator of an access request is the user (or subject), aiming to interact with a specific resource (or object). The resource represents the item or data the user intends to use or manipulate, while context refers to the surrounding conditions that influence whether access should be granted Operational circumstances under which access attempts take place are referred to as contexts. These situational factors, combined with other core elements, collectively shape the foundation of the HoBAC access control framework. In the HoBAC framework, each attribute consists of a distinctive label and an associated data element. The label serves as a unique reference to recognize the attribute, while the data element holds a specific value of a defined type. These

labels are linked to various system participants, enabling the formulation of access rules that rely on the corresponding attribute values. In the HoBAC system, permissions are governed by decision-making conditions that dictate approval or rejection of access attempts. These determinations rely on evaluating attributes tied to users, resources, and surrounding conditions. By analyzing these parameters against a defined collection of criteria, the system enforces authorization decisions. This attribute-driven approach enables the creation of adaptable and robust security configurations suitable for both IoT and traditional computing environments.

In the HoBAC model, combining attributes from various entities enables the formation of a unified subject or context with a more generalized representation. This method supports the definition of precise and adaptable access control strategies by allowing access requests to be consolidated into a single, abstracted form instead of handling them individually. Through this aggregation approach, the system minimizes exposure to individual-level components, reducing risks associated with unauthorized interactions. It also simplifies the administration and enforcement of access rules, streamlining identity and permission management while strengthening the overall security framework.

2.3 Implementation

The edge device was developed using the PyFUNS framework (Python Framework for Ubiquitous Networked Sensors), which is tailored for devices with limited resources. PyFUNS offers a simple and efficient API for creating IoT applications and supports ContikiOS, allowing seamless deployment across distributed environments and multiple platforms. The Chacha20 algorithm was employed to secure the communication through encryption. The protocol itself was coded in Python and executed on ContikiOS, an open-source operating system optimized for IoT applications. This platform was selected for its compatibility with resource-constrained devices and its minimal memory and processing requirements, making it well-suited for secure data exchange and authentication in IoT networks. Furthermore, ROSITA++ was utilized to detect and mitigate higher-order leakage, while ensuring the Chacha20 implementation remains secure through automatic masking. These strategies contribute to safe data transmission and safeguard against potential attacks, ensuring the protection of sensitive data within the network.

2.4 Security analysis of the proposed protocol

- Mutual authentication is a process used to confirm that communication is taking place between the correct parties. The shared value 'K' acts as a long-term key for this purpose, indicating that both entities have previously exchanged it. To maintain secure authentication, it is crucial to safeguard this key against unauthorized access and protect it from malicious threats

- Side-channel attacks target cryptographic systems by analyzing indirect data such as power usage, processing time, and electromagnetic emissions. To defend against these attacks, our proposed protocol incorporates the ROSITA tool to secure the masked version of Chacha, effectively preventing more than 99 % of potential data leaks from cryptographic processes. This significantly hampers attackers' ability to extract valuable information from the encrypted devices, thereby reinforcing the security of data transmission within IoT networks.

- To defend against Sybil attacks, our protocol employs robust authentication methods like two-factor authentication or biometrics to ensure the legitimacy of users. Furthermore, incorporating advanced access control strategies, such as Higher Order Attribute-based Access Control (HoBAC), along with secure encryption techniques, adds an extra layer of protection. Additionally, leveraging distributed trust networks and blockchain technology helps guarantee that only authenticated individuals can gain access to the network.

- To defend against replay attacks, the proposed protocol incorporates commitments and random numbers, which prevent attackers from retransmitting identical messages. The commitment mechanism ensures the validation of each step, while the use of random numbers generates distinct values for every session, making it challenging for an attacker to reuse previous messages. These strategies collectively enhance the security of data transmission and safeguard the system from malicious threats.

- A man-in-the-middle attack occurs when an attacker secretly intercepts and relays messages between two parties, who believe they are communicating directly with each other. Since the pre-shared key between the parties remains unknown to the attacker, it becomes extremely difficult for them to extract valuable information such as N1, N2, or ID_{Node} from the transmitted messages.

- Cryptanalysis attacks target the ciphertext, aiming to decipher the encryption by uncovering the key and revealing the plaintext. To safeguard against such attacks, a session key is employed, which is transmitted independently using a secure sub-algorithm, separate from the encrypted data.

- User anonymity: Suppose an attacker obtains a message represented as A sig_{pkuser}(N2,N1,ID_{Node}), H(K,N2)). However, extracting the ID_{Node} is extremely difficult because verifying the signature requires the user's private key, which remains confidential. Furthermore, obtaining the random numbers N1, N2, and the pre-shared key K is also highly challenging. Consequently, our protocol ensures robust protection of user anonymity.

- User untraceability: Imagine an attacker, A, intercepting two messages, A_{IDnode} and A'_{IDnode}, transmitted over an open channel, with the intent to trace the user. If any component of the two messages were identical, A could infer that both messages originated from the same user. However,

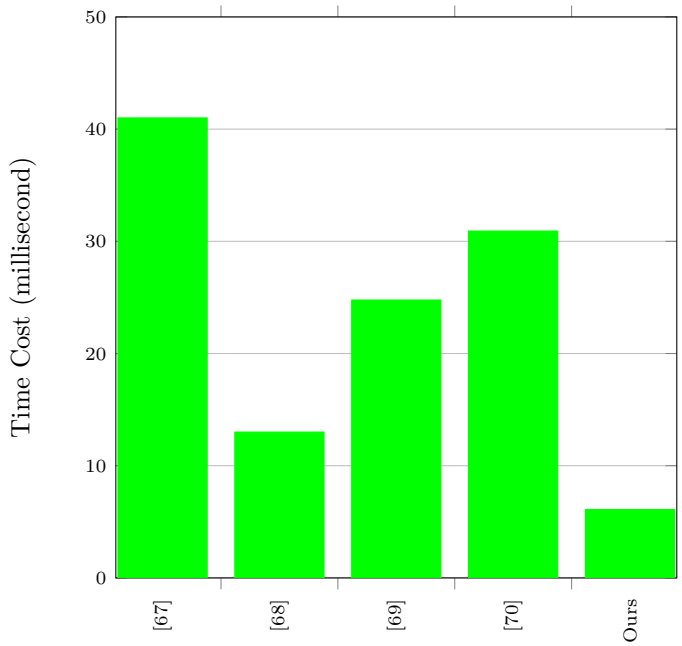

Figure 3.3: Time cost comparison

our protocol effectively prevents such tracing attempts because the values in A_{IDnode} and A'_{IDnode} are generated using random nonces, N1and N2. As a result, these values are unique to each session and remain valid only within that specific session.

2.4.1 Security proof with scyther

Scyther [66]emerges as a potent and efficient tool for the thorough examination and identification of potential security breaches and vulnerabilities within security protocols. Its automated analysis capabilities enable comprehensive scrutiny of protocol behavior, effectively assessing its resilience against a spectrum of potential attacks. One of Scyther's standout features, Niagree, offers assurance to communicating parties regarding the secure transmission and correct sequencing of messages, bolstering confidence in the integrity of data exchanges. Furthermore, the inclusion of the Alive feature serves to validate protocol steps, ensuring proper authorization by involved parties and mitigating the risk of unauthorized access. Scyther incorporates the Weakagree function as an essential safeguard to counteract identity spoofing, enhancing the overall protection of IoT infrastructures. By embedding this and other advanced capabilities, Scyther enables dependable and secure data exchange across connected devices, reinforcing the credibility and trustworthiness of communication within IoT ecosystems.

2.5 Performance evaluation

This part of the study analyzes how the suggested protocol performs by benchmarking it against existing methods, including those introduced by Shahidinejad et al. [67],Chen et al. [68], Chen et al. [69], and

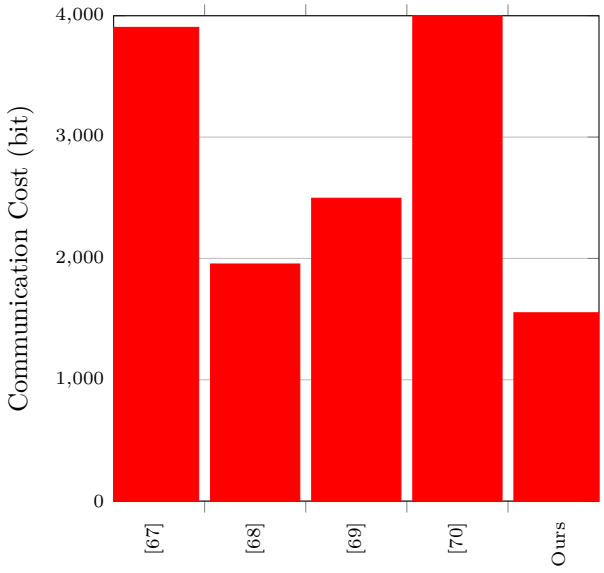

Figure 3.4: Communication cost comparison

Kumer and Om [70]. The comparison focuses on key efficiency indicators, specifically processing duration and data transmission overhead. Figure 3.3 presents a visual comparison of the execution durations across various protocols, including the one we developed. Analyzing the bar graph reveals that our method outperforms the alternatives in terms of speed, achieving the lowest computation time. Specifically, it surpasses the approaches detailed by Shahidinejad et al. [67], Chen et al. [68], Chen et al. [69], and Kumer and Om [70], which report times of approximately 13 ms, 41 ms, 24.77 ms, and 30.9166 ms, respectively.

In the comparison of communication costs, as illustrated in Figure 3.4, our proposed protocol outperforms other related protocols. The communication cost for our scheme is 1552 bits, while the counterpart protocol by Shahidinejad et al. [67]and Chen et al. [68], Chen et al. [69], kumer andOm [70], incurs a higher cost of 1954 bits, 3904 bits, 2496 bits and 4800 bits respectively. This disparity highlights the cost-effectiveness of our proposed scheme in communication expenditure, and it further emphasizes the comprehensive security features embedded in our solution, safeguarding against a spectrum of security attributes and potential threats.

3 lightweight authentication for iot edge devices

This study presents an efficient security mechanism aimed at authenticating IoT edge nodes. It is particularly suited for environments with numerous devices operating at the edge of the network, where bandwidth is constrained. The approach relies on streamlined symmetric encryption, utilizing the chacha20 cipher to manage session keys effectively. An in-depth verification using the Scyther analysis tool demonstrates that this method outperforms existing solutions, especially regarding latency and

data transmission efficiency.

3.1 The proposed authentication scheme

A three-phase architecture and protocol is introduced to enhance the security of IoT networks. The first phase involves registering edge nodes with a trusted server to establish their identity. The second phase focuses on authentication and the creation of a session key, ensuring the legitimacy of nodes and enabling secure communication. Finally, the session key is established between nodes to facilitate encrypted interactions. This protocol offers a robust solution for maintaining secure and dependable connectivity among IoT devices.

3.1.1 Architecture

This research introduces an innovative method to enhance IoT network security by utilizing an infrastructure based on edge computing concepts, illustrated in Figure 3.5.

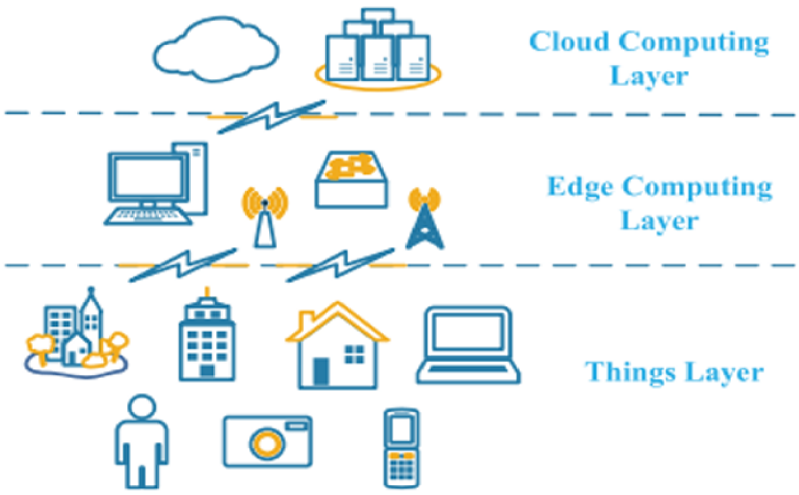

Figure 3.5: Architecture

The framework is built around smart IoT units positioned at the network's periphery, assigned to gather and transmit data. These devices are interconnected through a central edge node that coordinates their communication and ensures network cohesion . For processing tasks, the system employs a microcontroller known as PARAM to handle local computations efficiently. Central to our proposed protocol is the employment of the symmetric Chacha20 algorithm for data encryption purposes. By leveraging Chacha20, renowned for its robust security features and efficiency, we aim to furnish an effective and resilient solution for safeguarding sensitive data traversing IoT networks. The selection of Chacha20 as our encryption algorithm underscores our commitment to striking a balance between security and

performance, crucial considerations in the context of resource-constrained IoT environments. Through the integration of edge computing infrastructure and Chacha20 encryption, we envisage fortifying the security posture of IoT networks, thereby mitigating potential vulnerabilities and safeguarding against malicious threats.

3.1.2 The authentication protocol

In this section, we will present the phases of our authentication protocol :

- Edge node registration phase :
  - Blockchain technology enables devices, whether nodes or users, to obtain unique identifiers securely. This facilitates the safe exchange of data and provides a trusted platform for executing smart contracts. Furthermore, it preserves the anonymity of devices while leveraging the advantages of distributed ledger systems to their fullest potential.
  - Upon receiving the smart contract, the edge server securely stores it within a protected database. This approach safeguards the stored data against unauthorized access and malicious activities. To enhance security, the edge server implements multiple protective measures, including encryption, authentication, and access control, ensuring the integrity and confidentiality of the database.
- Authentication phase :

  The user first select a node and then requests its information.Our proposed authentication scheme works as follows:

  1) The user starts the session by requesting a random predefined offset time stamp (off_{TS}) from the server.
  2) The user creates a local timestamp variable named T_U, then transmits T_U, off_{TS}, and h (T_U,K) to the Node .
  3) When the Node receives the User's variables, it generates its local time stamp T_N.
  4) The node computes $T_{Noff} = T_N \oplus off_{TS}$; this number can be computed as a random number. and it is simple to generate and does notcost the node a lot of operation as well as the generation of random number.
  5) The node calculates the value of E1, which corresponds to its local ID and the value of E2 as specified below: $E_1 = h(ID_{\text{node}} \parallel T_U \parallel T_{Noff})$

     $E_2 = h(K, T_N)$.

     Subsequently, it returns to the user the values of E1, E2 and T_{Noff}.

6) Following the data reception from the previous step, the user proceeds by computing T'_N as $T_{Noff} \oplus off_{TS}$ to derive T_N. Then, it computes E3 where$E3 = h(T'_N, k)$.

7) Then it verifies if the difference $\Delta T = T_U - T'_N$ between its local time and the expected local time of the Node. If $\Delta T > 0$ and $E_3 \neq E_2$ then it aborts the protocol and it finds that (maybe) there is an attack.

8) The User computes $T_{Uoff} = T_U \oplus off_{TS}$ in order to hide its local time.

9) Then the receiver computes the value of E4, where $E4 = h(ID_{user} \parallel T_{Uoff})$ and sends the values of E1,E4, T_{Uoff}, T_{Noff}) and $h(T_{Uoff}, k)$to the database server.

10) At this phase, the server generates its local time stamp T_S.

11) The server checks if $\Delta' T = T_S - T'_u (= T_{Uoff} \oplus off_{TS})$ is positive, if yes, then it aborts the protocol and it concludes that there is an attack and the server it not authenticated neither the Node. Otherwise, it continues with the following steps:

    a) It Tries to find in its local records (Data) any User ID (ID'_{user}) that verifies the expression: $E_5 = h(ID'_{\text{user}} \parallel T_{Uoff})$. If $E_4 \not\equiv E_5$, then it concludes the User is not registered in the local database or there is an attack, in both cases the protocol is aborted. Otherwise, the User is authenticated.

    b) Then the server compute $E_6 = h\big(ID'_{\text{node}} \parallel (T_{Uoff} \oplus off_{TS}) \parallel T_{Noff}\big)$, and it starts the search in its local database to find a Node (ID'_{node}) which verifies the E6 by checking if $E_1 = E_6$, if it is not the case, then it concludes that the Node is not registered in the local database or there is an attack, in this case the protocol is aborted. Otherwise, the Node is authenticated successfully and it continues to the next steps.

    c) if $E_1 = E_6$, then the server computes the following expressions:
    $E_7 = h(ID'_{\text{user}} \parallel T_s)(1)$ $E_8 = h(ID'_{\text{node}} \parallel T_s)(2)$

    then, it sends the encryption message $\text{Enc}_k(T_s, E_7, E_8)$; its local time T_s and the values of E7 and E8 to the User.

12) The User receives the message ($Enc_k(T_s$, E7, E8)) from the server. After that, it computes the value of E9 , where $E_9 = (ID_{\text{user}} \parallel T_s)$ in order to verify if it is equal to the pre-computed one which received from the server (E7).

13) If $E_9 = E_7$, then the User verifies that it is authenticated by the database server, then it sends a message ($Enc_k(T_s, E_8)$) to the node. Otherwise, it aborts the protocol.

14) After the user is authenticated, the Node receives the time stamp of the server and the value of E8 from the server. At this phase, the node calculates the value of E10, where $E_{10} = h(ID_{\text{node}} \parallel T_S)$

15) If $E_{10} = E_8$ then the Node is authenticated successfully and verifies that it is authenticated by the Database server. Otherwise, it aborts the protocol.

► Establish session key phase :

Using the FTKD for Identity-based Threshold Symmetric Encryption, compute the message specific whole key wk_i to send the session key $(k \parallel T_S \parallel off_{TS})$, $wk_i = \mathrm{e}(H_1(id_{\mathrm{Node}})), \quad H_2(\mathrm{com}(k \parallel T_S \parallel off_{TS}, v_i))^s$,all the users will be able to connect to the same node by using the node's identity. Then, the node will access to any message encrypted under its identity[19].

3.2 Security analysis of the proposed protocol

We examine the security characteristics in this section. Table 3-2 presents the results. In the table, we use a checkmark (✓) to indicate that the scheme possesses this security property. Otherwise, a cross (×) is used. The comparative analysis demonstrates that the proposed scheme offers superior security compared to the existing authentication scheme in the IoT network as illustrated in Table 3-2.

Table 3.2: Security properties analysis

	[71]	[72]	[73]	[74]	[75]	[67]	[68]	Ours
Mutual authentication	✓	✓	✓	✓		✓	✓	✓
Replay attack	✓	✓	✓	✓		×	✓	✓
Perfect forward secrecy	✓	✓	×	×	×	×	✓	✓
MITM	✓	×	×	✓		✓	✓	✓
Side channel attack	×	×	×	×	×	×	×	✓

- Mutual authentication verifies that both communicating parties are interacting with their intended counterparts. Since K serves as a long-term shared key for mutual authentication, it is assumed that the key has already been exchanged between the entities. Additionally, the proposed protocol incorporates a dual-layer authentication process, which effectively mitigates the risk of such attacks.

- Protection against replay attacks is achieved in the proposed protocol through the use of commitments and synchronized local timestamps. Commitments ensure that every step in the process is validated, while each new connection relies on the local time of the node along with a timestamp offset generated by the server, the user, and the server. Additionally, an offset value is incorporated to enhance overall security. These measures collectively safeguard data transmission and provide robust protection against potential malicious activities.

- Side-channel attacks are a method used to analyze cryptographic algorithms by exploiting information such as power consumption, execution time, and electromagnetic emissions. To counteract power-based side-channel attacks, the proposed solution employs a microprocessor with integrated security features, referred to as PARAM. This design significantly hinders attackers from extracting

valuable data from encryption devices, thereby enhancing the security of data transmission within IoT networks.

- Perfect Forward Secrecy guarantees that past communications remain secure even if a long term secret key is compromised. This means that, even if an attacker obtains the private encryption keys used in previous interactions, they would still be unable to decrypt the earlier communication data. In the proposed protocol, even if the long-term secret key (pre-shared key k) or other elements used to generate the session key are exposed, it is impossible for an attacker to retrieve past session keys. As a result, the proposed protocol successfully achieves Perfect Forward Secrecy.

- man-in-the-middle attack occurs when an attacker covertly intercepts and manipulates communications between two parties who assume they are directly connected. Utilizing a pre shared key between the two entities effectively mitigates this type of attack by enabling secure mutual authentication before initiating communication. Moreover, it becomes challenging for an attacker to extract sensitive information, such as TN , IDnode, from the transmitted messages. Additionally, the attacker cannot alter the variable values exchanged between the user and the server or the messages forwarded between the server, the user, and the node.

3.2.1 Security Proof With Scyther

The tool known as Scyther [66] provides a powerful automated method for analyzing security protocol designs. It is specifically designed to detect weaknesses and expose possible exploit paths by simulating various attack scenarios, allowing for detailed evaluation of a protocol's defense mechanisms. Among Scyther's notable capabilities is a function called Niagree, which helps ensure that messages are both securely delivered and correctly ordered, enhancing trust in communication integrity. The tool also incorporates the Alive check, which confirms that all steps in the protocol are performed by authorized entities, reducing the risk of unauthorized actions. Additionally, the Weakagree function offers protection against identity spoofing, reinforcing defenses against impersonation threats. Together, these sophisticated features enable Scyther to support the development of trustworthy and secure data exchanges in IoT systems, ensuring dependable and protected communication between devices.

As illustrated in Figure 3.6, the evaluation performed using Scyther validates the robustness of the security mechanisms embedded in our designed framework. The results clearly demonstrate the system's strong resistance to known attack vectors. In essence, Scyther played a critical role in our study by enabling a detailed inspection of the authentication protocol's security requirements, identifying potential flaws, and reinforcing the system's overall protection strategy.

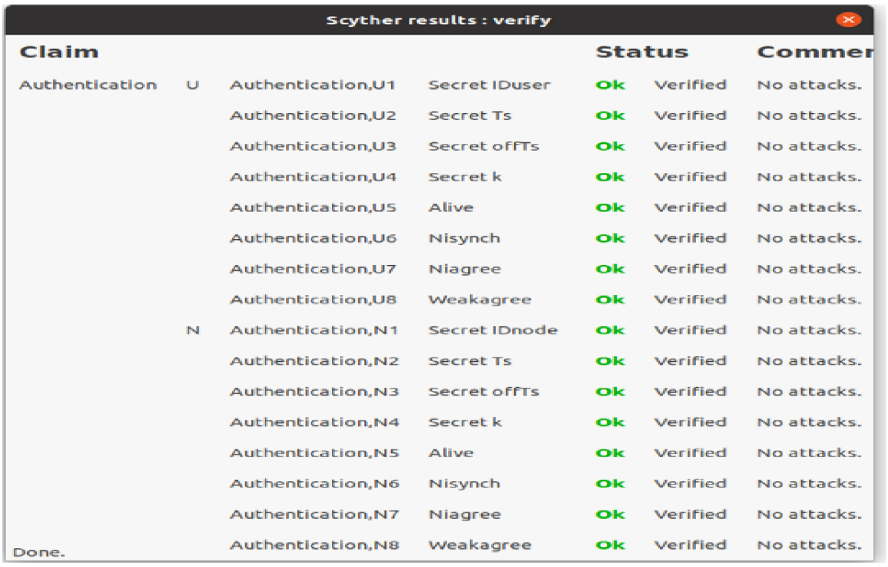

Figure 3.6: Scyther simulation outcames of our protocol

3.3 Performance evaluation

In this section, we conduct a comprehensive examination and comparative analysis of various edge computing protocols, juxtaposed against our proposed protocol, to highlight its superior performance characteristics.

Figure 3.7 provides a detailed visual comparison of the execution durations for several protocols, including the one we introduced. The graphical representation highlights a noticeable performance advantage of our method. Specifically, our protocol achieves a processing time of just 11.5 milliseconds, making it substantially faster than those outlined in studies [67] and [68], which require roughly 13 and 41 milliseconds, respectively. This clearly underscores the superior efficiency of our solution.

As shown in Figure 3.8, our designed protocol achieves notable communication efficiency compared to similar existing methods. It requires only 1472 bits for data exchange 256 bits allocated for Chacha20-based encryption and decryption, and 160 bits assigned to the cryptographic hash function. In comparison, the protocols introduced by Chen et al. [68] and Ali et al. [67] demand significantly more, consuming 3904 bits and 1954 bits, respectively.

This notable reduction in communication overhead highlights the lightweight and optimized nature of our solution, making it particularly well-suited for edge computing environments.

Additionally, the lower data transmission requirement does not compromise the system's robustness. Instead, it reflects a careful integration of security mechanisms that offer strong protection while conserving bandwidth and computational resources.

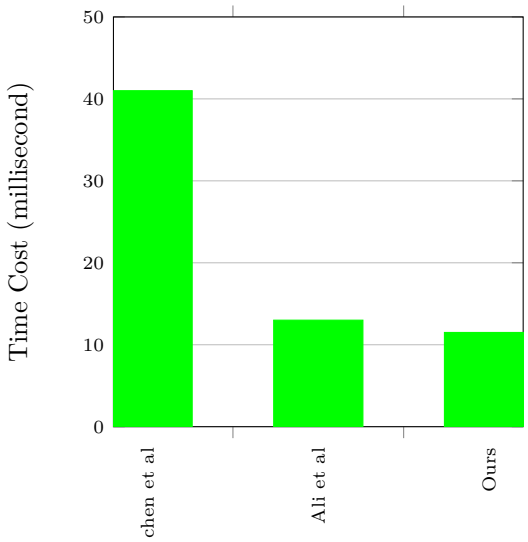

Figure 3.7: Time cost comparison

4 Conclusion

In this chapter, we presented our contribution through lightweight authentication schemes for edge networks in the context of the Internet of Things. The evaluation results showed that they are very lightweight and well-suited for such an environment. These authentication schemes are highly effective when the communicating objects have very limited computing and storage resources.

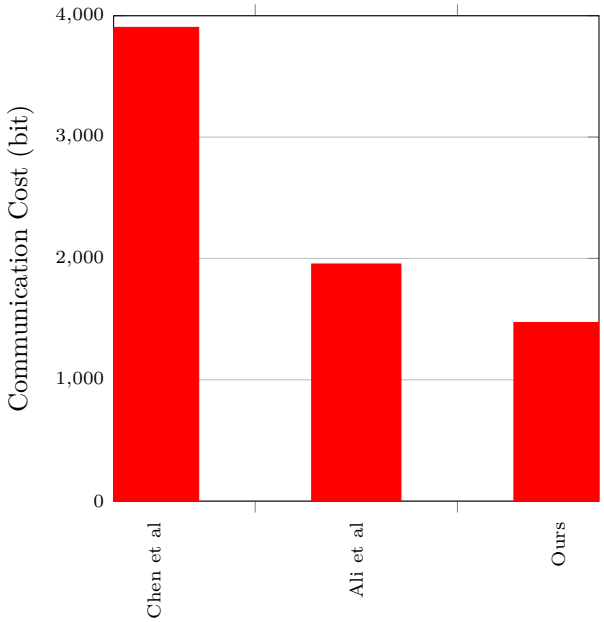

Figure 3.8: Communication cost comparison

General conclusion

Managing authentication in IoT is crucial for maintaining the security and reliability of various applications. These applications are highly susceptible to threats such as identity theft and side channel attacks, which can compromise their functionality. To mitigate these risks, identity authentication and protection must be implemented in a secure and efficient manner. The primary objective of authentication mechanisms is to verify and confirm the identity of each connected device. However, deploying such mechanisms in an IoT environment requires careful consideration of resource limitations, including computing power and storage capacity.

In this thesis, we explored the IoT landscape, including its diverse application domains and the key security challenges it faces. We provided a comprehensive review of identity management and examined various authentication protocols designed for IoT environments. Furthermore, we presented our contributions, which constitute the foundation of our research.

We presented two lightweight schemes. In the first one , we presented an efficient lightweight authentication and access control for iot edge devices . This method has been developed to manage authentication and permissions specifically for Internet of Things (IoT) edge environments. This strategy supports a wide range of edge devices and allows for efficient data transfer despite bandwidth limitations by leveraging a lightweight symmetric encryption scheme ;specifically, the ChaCha20 cipher for secure session key setup. Furthermore, it strengthens identity verification and access control through the integration of blockchain frameworks and the HoBAC mechanism.

The second one, lightweight authentication for iot edge devices . This protocol alow an edge device and edge server to authenticate each auther mutually to ensure the security of the collected data . It used the hash function and the symetric encryption to tramsmit the date during the authentication process. The analysis of the proposed schemes in terms of security and performance shows that they offer computational and storage efficiency, while maintaining resistance to various possible attacks and ensuring security properties.

As perspectives for our research work, we have new directions focused on novel identity authentication and identity protection schemes. The implementation of our proposed schemes in a real environment is also planned in order to obtain more analytical results regarding security and performance.

- We aim to explore the integration of continuous authentication mechanisms within IoT environments, particularly in edge networks. Unlike traditional authentication, continuous authentication can provide real-time identity verification by leveraging behavioral biometrics, contextual data, and machine learning. This approach has the potential to enhance security without compromising user experience, especially in dynamic and resource-constrained environments. Our future work will focus on designing lightweight, privacy-preserving continuous authentication schemes adapted to

the constraints of IoT devices.

- We will propose leveraging capability-based access control (CapBAC) for decentralized IoT networks to reduce reliance on central authorities. By issuing cryptographic tokens representing fine-grained permissions, CapBAC enables secure peer-to-peer authorization while supporting scalability and autonomy at the edge. We will evaluates the feasibility of Zero Trust Architecture (ZTA) in edge IoT systems, where no device or user is implicitly trusted. Implementing ZTA principles can mitigate lateral movement in compromised networks and improve resilience against insider threats.

- We are investigating hybrid access control models that combine role-, attribute-, and capability-based techniques for flexible and fine-grained policy enforcement. This hybrid approach aims to balance security, expressiveness, and computational efficiency for diverse edge IoT scenarios.

- Our future work includes designing privacy-preserving access control schemes using homomorphic encryption and secure multiparty computation. These methods allow sensitive access rules and decisions to be enforced without revealing underlying data, preserving confidentiality even in collaborative IoT ecosystems.

List of publication

International journal

1

Authors: Imane Zerraza, Zianou Ahmed Seghir, Mounir Hemam

Title: An Efficient Lightweight Authentication and Access Control for IoT Edge Devices.

Journal: International Journal of Safety and Security Engineering

Volume: 14

Issue Numbers: 3

Pages: 807-813

link: https://doi.org/10.18280/ijsse.140313

2

Authors: Imane Zerraza

Title: Lightweight Authentication for IoT Edge Devices.

Journal: informatica

Volume: 48

Issue Numbers: 18

Pages: 15-20

link: https://doi.org/10.31449/inf.v48i18.6012

Bibliography

[1] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

[2] Hadi Habibzadeh, Tolga Soyata, Burak Kantarci, Azzedine Boukerche, and Cem Kaptan. Sensing, communication and security planes: A new challenge for a smart city system design. *Computer Networks*, 144:163–200, 2018.

[3] Mark Weiser, Rich Gold, and John Seely Brown. The origins of ubiquitous computing research at parc in the late 1980s. *IBM systems journal*, 38(4):693–696, 1999.

[4] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

[5] Mirko Presser, Payam M Barnaghi, Markus Eurich, and Claudia Villalonga. The sensei project: Integrating the physical world with the digital world of the network of the future. *IEEE Communications Magazine*, 47(4):1–4, 2009.

[6] ZigBee Alliance. Zigbee alliance web page. `http://www.zigbee.org/`.

[7] Usman Raza, Parag Kulkarni, and Mahesh Sooriyabandara. Low power wide area networks: An overview. *ieee communications surveys & tutorials*, 19(2):855–873, 2017.

[8] Marco Centenaro, Lorenzo Vangelista, Andrea Zanella, and Michele Zorzi. Long-range communications in unlicensed bands: The rising stars in the iot and smart city scenarios. *IEEE Wireless Communications*, 23(5):60–67, 2016.

[9] Sigfox Corporation. Sigfox geolocation. `https://www.sigfox.com/en/sigfox-geolocation/`.Accessed: 20/01/2025.

[10] Tomasz Kobialka, Rajkumar Buyya, Peng Deng, Lars Kulik, and Marimuthu Palaniswami. Sensor web: Integration of sensor networks with web and cyber infrastructure. In *Handbook of research on developments and trends in wireless sensor networks: From principle to practice*, pages 447–473. IGI Global Scientific Publishing, 2010.

[11] Hadi Habibzadeh, Karthik Dinesh, Omid Rajabi Shishvan, Andrew Boggio-Dandry, Gaurav Sharma, and Tolga Soyata. A survey of healthcare internet of things (hiot): A clinical perspective. *IEEE internet of things journal*, 7(1):53–71, 2019.

[12] Hande Alemdar and Cem Ersoy. Wireless sensor networks for healthcare: A survey. *Computer networks*, 54(15):2688–2710, 2010.

[13] Fadele Ayotunde Alaba, Mazliza Othman, Ibrahim Abaker Targio Hashem, and Faiz Alotaibi. Internet of things security: A survey. *Journal of network and computer applications*, 88:10–28, 2017.

[14] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646, 2016.

[15] Susha Surendran, Amira Nassef, and Babak D Beheshti. A survey of cryptographic algorithms for iot devices. In *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pages 1–8. IEEE, 2018.

[16] Daniel J Bernstein et al. Chacha, a variant of salsa20. In *Workshop record of SASC*, pages 3–5. Lausanne, Switzerland, 2008.

[17] Fabrizio De Santis, Andreas Schauer, and Georg Sigl. Chacha20-poly1305 authenticated encryption for high-speed embedded iot applications. In *Design, automation & test in europe conference & exhibition (DATE), 2017*, pages 692–697. IEEE, 2017.

[18] Jan Pelzl. *Understanding cryptography: a textbook for students and practitioners*. Springer, 2010.

[19] David McGrew, Kevin Igoe, and Margaret Salter. Fundamental elliptic curve cryptography algorithms, 2011.

[20] Alex Biryukov and Leo Perrin. State of the art in lightweight symmetric cryptography. *Cryptology ePrint Archive*, 2017.

[21] Gregor Leander, Christof Paar, Axel Poschmann, and Kai Schramm. New lightweight des variants. In *International workshop on fast software encryption*, pages 196–210. Springer, 2007.

[22] Christophe De Canniere, Orr Dunkelman, and Miroslav Knežević. Katan and ktantan—a family of small and efficient hardware-oriented block ciphers. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 272–288. Springer, 2009.

[23] Andrey Bogdanov, Lars R Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew JB Robshaw, Yannick Seurin, and Charlotte Vikkelsoe. Present: An ultra-lightweight block cipher. In *International workshop on cryptographic hardware and embedded systems*, pages 450–466. Springer, 2007.

[24] Daniel Engels, Markku-Juhani O Saarinen, Peter Schweitzer, and Eric M Smith. The hummingbird-2 lightweight authenticated encryption algorithm. In *International workshop on radio frequency identification: Security and privacy issues*, pages 19–31. Springer, 2011.

[25] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK families of lightweight block ciphers. Cryptology ePrint Archive, Paper 2013/404, 2013.

[26] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matt Robshaw. The led block cipher. In *International workshop on cryptographic hardware and embedded systems*, pages 326–341. Springer, 2011.

[27] François-Xavier Standaert, Gilles Piret, Neil Gershenfeld, and Jean-Jacques Quisquater. Sea: A scalable encryption algorithm for small embedded applications. In *International conference on smart card research and advanced applications*, pages 222–236. Springer, 2006.

[28] Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. : a lightweight block cipher for multiple platforms. In *International Conference on Selected Areas in Cryptography*, pages 339–354. Springer, 2012.

[29] NIST Skipjack. Kea algorithm specifications. *Online document: http://csrc. nist. org/encryption/skipjack/skipjack. pdf*, 1998.

[30] Joan Daemen, Michaël Peeters, Gilles Van Assche, and Vincent Rijmen. Nessie proposal: Noekeon. In *First open NESSIE workshop*, pages 213–230, 2000.

[31] Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bon-Seok Koo, Changhoon Lee, Donghoon Chang, Jesang Lee, Kitae Jeong, et al. Hight: A new block cipher suitable for low-resource device. In *International workshop on cryptographic hardware and embedded systems*, pages 46–59. Springer, 2006.

[32] Chae Hoon Lim and Tymur Korkishko. mcrypton–a lightweight block cipher for security of low-cost rfid tags and sensors. In *International workshop on information security applications*, pages 243–258. Springer, 2005.

[33] Xiaoyang Zhu and Youakim Badr. Identity management systems for the internet of things: a survey towards blockchain solutions. *Sensors*, 18(12):4215, 2018.

[34] Telecommunication Standardization Sector. Ngn identity management framework. Available online: https://www.itu.int/rec/T-REC-Y.2720-200901-I (accessed on 31/01/2025).

[35] Audun Jøsang and Simon Pope. User centric identity management. In *AusCERT Asia Pacific information technology security conference*, volume 22, page 2005. Brisbane, QLD, 2005.

[36] PrimeLife. Primelife-privacy and identity management in europe for life. *online] http://www. primelife. eu/(accessed 20 september 2024)*.

[37] SWIFT. Secure widespread identities for federated telecommunications. Available online: `https://www.swift.com` (accessed on 20/09/2024).

[38] DAIDALOS. Designing advanced network interfaces for the delivery and administration of location independent, optimised personal services. Available online: `https://www.tssg.org/projects/daidalos/` (accessed on 20/09/2024).

[39] Kantara. Available online: `http://kantarainitiative.org` (accessed on 20/09/2024).

[40] Liberty Identity Assurance Framework. Liberty alliance project, 2007.

[41] Vashek Matyáš, Simone Fischer-Hübner, Daniel Cvrcek, and Petr Švenda. *The Future of Identity in the Information Society: 4th IFIP WG 9.2, 9.6, 11.6, 11.7/FIDIS International Summer School, Brno, Czech Republic, September 1-7, 2008, Revised Selected Papers*, volume 298. Springer Science & Business Media, 2009.

[42] OASIS Security Services Technical Committee. Security assertion markup language (saml) v2.0. Available online: `https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security` (accessed on 20/09/2024).

[43] Higgins Project. Open source identity framework. Available online: `https://www.eclipse.org/higgins` (accessed on 20/09/2024).

[44] David Recordon and Drummond Reed. Openid 2.0: a platform for user-centric identity management. In *Proceedings of the second ACM workshop on Digital identity management*, pages 11–16, 2006.

[45] Scott Cantor and Tom Scavo. Shibboleth architecture. *Protocols and Profiles*, 10(16):29, 2005.

[46] STORK Project. Secureidentity across borders linked (stork). Available online: `https://www.eid-stork.eu` (accessed on 20/09/2024).

[47] PICOS Project. Privacy and identity management for community services. Available online: `http://www.picos-project.eu` (accessed on 20/09/2024).

[48] Vittorio Bertocci, Garrett Serack, and Caleb Baker. *Understanding windows cardspace: an introduction to the concepts and challenges of digital identities*. Pearson Education, 2007.

[49] Satoshi Nakamoto and A Bitcoin. A peer-to-peer electronic cash system. *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, 4(2):15, 2008.

[50] Matthew Beedham. All you need to know about bitcoin network nodes. *The Next Web*, 1, 2019.

[51] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. Blockchain challenges and opportunities: A survey. *International journal of web and grid services*, 14(4):352–375, 2018.

[52] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2009.

[53] Arati Baliga. Understanding blockchain consensus models. *Persistent*, 4(1):14, 2017.

[54] BitShares. Bitshares 2.0: Distributed consensus. Technical report, BitShares Whitepapers, 2015.

[55] Ethereum Foundation. Ethereum: A next-generation smart contract and decentralized application platform. Available online: `https://github.com/Ethereum/wiki/wiki/White-Paper` (accessed on 02/12/2024).

[56] Bhabendu Kumar Mohanta, Soumyashree S Panda, and Debasish Jena. An overview of smart contract and use cases in blockchain technology. In *2018 9th international conference on computing, communication and networking technologies (ICCCNT)*, pages 1–4. IEEE, 2018.

[57] Christian Cachin. Architecture of the hyperledger blockchain fabric, 2016.

[58] Sanaah Al Salami, Joonsang Baek, Khaled Salah, and Ernesto Damiani. Lightweight encryption for smart home. In *2016 11th International conference on availability, reliability and security (ARES)*, pages 382–388. IEEE, 2016.

[59] Freddy K Santoso and Nicholas CH Vun. Securing iot for smart home system. In *2015 international symposium on consumer electronics (ISCE)*, pages 1–2. IEEE, 2015.

[60] Ola Salman, Sarah Abdallah, Imad H Elhajj, Ali Chehab, and Ayman Kayssi. Identity-based authentication scheme for the internet of things. In *2016 IEEE Symposium on Computers and Communication (ISCC)*, pages 1109–1111. IEEE, 2016.

[61] Ning Ye, Yan Zhu, Ru-chuan Wang, Reza Malekian, and Qiao-min Lin. An efficient authentication and access control scheme for perception layer of internet of things, 2014.

[62] Adriano Witkovski, Altair Santin, Vilmar Abreu, and João Marynowski. An idm and key-based authentication method for providing single sign-on in iot. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2015.

[63] SM Salim Reza, Afida Ayob, Md Murshedul Arifeen, Nowshad Amin, Mohd Hanif Md Saad, and Aini Hussain. A lightweight security scheme for advanced metering infrastructures in smart grid. *Bulletin of Electrical Engineering and Informatics*, 9(2):777–784, 2020.

[64] Imane Zerraza, Zianou Ahmed Seghir, and Mounir Hemam. An efficient lightweight authentication and access control for iot edge devices. *International Journal of Safety & Security Engineering*, 14(3), 2024.

[65] Imane Zerraza. Lightweight authentication for iot edge devices. *Informatica*, 48(18), 2024.

[66] CISPA. Scyther protocol analysis tool. Available online: `https://people.cispa.io/cas.cremers/scyther/` (accessed on 16/03/2024).

[67] Ali Shahidinejad, Mostafa Ghobaei-Arani, Alireza Souri, Mohammad Shojafar, and Saru Kumari. Light-edge: A lightweight authentication protocol for iot devices in an edge-cloud environment. *IEEE consumer electronics magazine*, 11(2):57–63, 2021.

[68] Chien-Ming Chen, Lili Chen, Yanyu Huang, Sachin Kumar, and Jimmy Ming-Tai Wu. Lightweight authentication protocol in edge-based smart grid environment. *EURASIP Journal on Wireless Communications and Networking*, 2021(1):68, 2021.

[69] Chien-Ming Chen, Zhaoting Chen, Saru Kumari, and Meng-Chang Lin. Lap-ioht: A lightweight authentication protocol for the internet of health things. *Sensors*, 22(14):5401, 2022.

[70] Pankaj Kumar and Hari Om. Multi-ta model-based conditional privacy-preserving authentication protocol for fog-enabled vanet. *Vehicular Communications*, 47:100785, 2024.

[71] Pankaj Kumar and Lokesh Chouhan. A secure authentication scheme for iot application in smart home. *Peer-to-Peer Networking and Applications*, 14(1):420–438, 2021.

[72] Fan Wu, Xiong Li, Lili Xu, Saru Kumari, Marimuthu Karuppiah, and Jian Shen. A lightweight and privacy-preserving mutual authentication scheme for wearable devices assisted by cloud server. *Computers & Electrical Engineering*, 63:168–181, 2017.

[73] Umair Khalid, Muhammad Asim, Thar Baker, Patrick CK Hung, Muhammad Adnan Tariq, and Laura Rafferty. A decentralized lightweight blockchain-based authentication mechanism for iot systems. *Cluster Computing*, 23(3):2067–2087, 2020.

[74] Nan Li, Dongxi Liu, and Surya Nepal. Lightweight mutual authentication for iot and its applications. *IEEE Transactions on Sustainable Computing*, 2(4):359–370, 2017.

[75] Ali S Rachini and R Khatoun. Distributed key management authentication algorithm in internet of things (iot). In *2020 Sixth International Conference on Mobile And Secure Services (MobiSecServ)*, pages 1–5. IEEE, 2020.