People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Echahid Cheikh Larbi Tebessi University - Tebessa

Faculty of Exact Sciences, Natural and Life Sciences
Department of Computer Science / Mathematics
Domiciliation laboratory: LAboratory of Mathematics, Informatics and Systems (LAMIS)

THESIS

Presented and Publicly Defended in Partition for the Award of a Doctoral Degree

by:
Amira Cheriet

Field: Mathematics and Computer Science. **Field of study:** Informatics/ Mathematics
Speciality: Information System

Titled

Analysis and improvement of QoS and security of Mobile Edge Computing based networks

Defended on : 11/05/2026 Before the jury composed of :

Full name	Grade		
Mr. Hakim Bendjenna	Pr	Univ. of Echahid Cheikh Larbi Tebessi	President
Mr. Tahar Mekhaznia	Pr	Univ. of Echahid Cheikh Larbi Tebessi	Supervisor
Mr. Makhlouf DERDOUR	Pr	Univ. of Oum el Bouaghi	Examiner
Mr. Zakaria Benmounah	MCA	Univ. of Constantine 2	Examiner
Mr. Gasmi Mohamed	MCA	Univ. of Echahid Cheikh Larbi Tebessi	Examiner

Academic Year : 2025/2026

Analysis and improvement of QoS and security of Mobile Edge Computing based networks

Cheriet Amira

Echahid Cheikh Larbi Tebessi University

Laboratory of Mathematics, Informatics and Systems (LAMIS)

Résumé

L'évolution rapide des technologies mobiles et des objets connectés (IoT) a entraîné une augmentation massive du volume de données à traiter et à sécuriser. Les architectures cloud traditionnelles, bien qu'efficaces pour le stockage et le traitement à grande échelle, ne répondent plus aux exigences actuelles en termes de latence, de bande passante et de sécurité. Pour répondre à ces défis, le paradigme du Mobile Edge Computing (MEC) s'impose comme une solution innovante permettant de rapprocher les ressources de calcul et de stockage des utilisateurs finaux, tout en améliorant les performances des services distribués.

Cette thèse propose une série de contributions visant à optimiser le traitement, la sécurité et le stockage des données sensibles dans des environnements MEC, à travers l'intégration de technologies complémentaires comme le Software-Defined Networking (SDN), la Blockchain, le Network Functions Virtualization (NFV) et le système IPFS.

Dans un premier volet, nous concevons une architecture sécurisée d'offloading de données mobiles dans des réseaux de petites cellules (SCN), combinant SDN, blockchain publique/privée et communications Wi-Fi et D2D. Cette solution vise à garantir une latence ultra-faible tout en assurant l'intégrité et la confidentialité des données transférées depuis les terminaux mobiles vers le cloud.

Dans un second temps, nous abordons la problématique de la gestion sécurisée des flux de streaming vidéo en périphérie. Nous proposons un modèle distribué basé sur l'orchestration dynamique des tâches de sécurité via NFV, permettant d'optimiser l'allocation des ressources tout en assurant une qualité de service constante, même sous contrainte de performance.

Enfin, nous présentons une solution de stockage sécurisé pour les systèmes de vidéo surveillance, basée sur la blockchain et l'IPFS, avec une couche MEC permettant le prétraitement local. L'utilisation de mécanismes comme le Segregated Witness et les smart contracts assure l'intégrité, la traçabilité et l'efficience du stockage des données vidéos critiques.

L'ensemble des propositions est validé par des simulations et des évaluations expérimentales qui démontrent leur efficacité en termes de réduction de la latence, d'amélioration de la sécurité, de résilience des données, et d'optimisation des ressources. Cette thèse ouvre ainsi la voie vers des environnements edge intelligents, sécurisés, and performants, adaptés to croissant needs des applications modernes en mobilité, en particulier in the domaines of l'IoT, des services multimédias, and of the security urbaine.

Mots Clés: Mobile Edge Computing, Video Security, Blockchain, SDN, QoS, IPFS.

Abstract

The rapid evolution of mobile technologies and connected objects (IoT) has led to a massive increase in the volume of data to be processed and secured. Traditional cloud architectures, while efficient for large-scale storage and processing, no longer meet current requirements in terms of latency, bandwidth, and security. To address these challenges, the Mobile Edge Computing (MEC) paradigm is emerging as an innovative solution that brings computing and storage resources closer to end users while improving the performance of distributed services.

This thesis offers a series of contributions aimed at optimizing the processing, security, and storage of sensitive data in MEC environments through the integration of complementary technologies such as Software-Defined Networking (SDN), Blockchain, Network Functions Virtualization (NFV), and IPFS.

In the first part, we design a secure mobile data offloading architecture in small cell networks (SCN), combining SDN, public/private blockchain, and Wi-Fi and D2D communications. This solution aims to guarantee ultra-low latency while ensuring the integrity and confidentiality of data transferred from mobile devices to the cloud.

Secondly, we address the issue of secure edge video streaming management. We propose a distributed model based on dynamic orchestration of security tasks via NFV, allowing for optimized resource allocation while ensuring consistent quality of service, even under performance constraints.

Finally, we present a secure storage solution for video surveillance systems, based on blockchain and IPFS, with a MEC layer enabling local preprocessing. The use of mechanisms such as Segregated Witness and smart contracts ensures the integrity, traceability, and efficient storage of critical video data.

All the proposals are validated by simulations and experimental evaluations that demonstrate their effectiveness in terms of latency reduction, security improvement, data resilience, and resource optimization. This thesis thus paves the way towards intelligent, secure, and efficient edge environments, adapted to the growing needs of modern mobile applications, particularly in the fields of IoT, multimedia services, and urban security.

Keywords: Mobile Edge Computing, Video Security, Blockchain, SDN, QoS, IPFS.

الملخص

أدى التطور السريع لتقنيات الأجهزة المحمولة والأجهزة المتصلة (إنترنت الأشياء) إلى زيادة هائلة في حجم البيانات المطلوب معالجتها وتأمينها. ورغم كفاءة هياكل السحابة التقليدية في التخزين والمعالجة واسعة النطاق، إلا أنها لم تعد تلبي المتطلبات الحالية من حيث زمن الوصول وعرض النطاق الترددي والأمان. ولمواجهة هذه التحديات، يبرز نموذج الحوسبة الطرفية المتنقلة (MEC) كحل مبتكر يُقرّب موارد الحوسبة والتخزين من المستخدمين النهائيين، مع تحسين أداء الخدمات الموزعة.

تقدم هذه الأطروحة سلسلة من المساهمات التي تهدف إلى تحسين معالجة البيانات الحساسة وأمنها وتخزينها في بيئات الحوسبة الطرفية المتنقلة، من خلال دمج تقنيات تكميلية مثل الشبكات المعرفة بالبرمجيات (SDN)، وسلسلة الكتل (Blockchain)، ومحاكاة وظائف الشبكة الافتراضية (NFV)، ونظام الملفات IPFS. في الجزء الأول، نصمم هيكلية آمنة لتفريغ البيانات المتنقلة في شبكات الخلايا الصغيرة (SCN)، تجمع بين SDN، وسلسلة الكتل العامة/الخاصة، واتصالات Wi-Fi وD2D. يهدف هذا الحل إلى ضمان زمن وصول منخفض للغاية مع ضمان سلامة وسرية البيانات المنقولة من الأجهزة المحمولة إلى السحابة. ثانيًا، نتناول مسألة إدارة بث الفيديو الطرفي الآمن. نقترح نموذجًا موزعًا قائمًا على التنسيق الديناميكي لمهام الأمان عبر NFV، مما يسمح بتخصيص الموارد على نحوٍ مُحسَّن مع ضمان جودة خدمة ثابتة، حتى في ظل قيود الأداء. وأخيرًا، نقدم حلًا آمنًا لتخزين أنظمة المراقبة بالفيديو، قائمًا على تقنية البلوك تشين (blockchain) ونظام الملفات متعدد البروتوكولات (IPFS)، مع طبقة MEC تُمكّن من المعالجة المسبقة المحلية. يضمن استخدام آليات مثل Witness Segregated والعقود الذكية سلامة بيانات الفيديو المهمة وإمكانية تتبعها وتخزينها بكفاءة. تم التحقق من صحة جميع المقترحات من خلال عمليات محاكاة وتقييمات تجريبية تُثبت فعاليتها من حيث تقليل زمن الوصول، وتحسين الأمان، ومرونة البيانات، وتحسين الموارد. وبالتالي، تمهد هذه الأطروحة الطريق نحو بيئات طرفية ذكية وآمنة وفعالة، تتكيف مع الاحتياجات المتزايدة لتطبيقات الهاتف المحمول الحديثة، لا سيما في مجالات إنترنت الأشياء (IoT)، وخدمات الوسائط المتعددة، والأمن الحضري.

الكلمات المفتاحية: الحوسبة المتنقلة على الحافة، أمان الفيديو، البلوك تشين، SDN، جودة الخدمة، IPFS.

Dedication

To my husband and my little girl Watine,
To my father,
To my sisters.

Acknowledgment

I am among those who firmly believe that there is no power or strength except through God. My first thanks go to Almighty God for the will, health, and patience He has granted me throughout these years of study.

This thesis could not have been completed without the support and assistance of many people. I would like to take this opportunity to express my deep gratitude to all those who contributed to this work, whether colleagues, family, or friends.

I would like to warmly thank the jury members for accepting to review this thesis and for their constructive feedback. I am deeply grateful for the attention you have given to my manuscript and the valuable time you have dedicated to its evaluation.

I am especially grateful to my in-laws, as well as to my father and sisters, for their constant interest in the success of my studies and their support, often far more valuable than they might realize.

I would also like to express my heartfelt thanks to my thesis supervisor, Dr. Mekhaznia Tahar, for agreeing to oversee my work. He introduced me to a new field of research, Mobile Edge Computing. His insightful ideas and guidance helped enrich and refine my initial proposals.

A big thank you as well to my university colleagues for their support and camaraderie.

Finally, I want to express my deep gratitude to my husband, Dr. Sahraoui Abdelatif, for his unconditional support, and to my little daughter, Watine, who brought me joy and inspiration throughout this journey.

Thank you all!

Table of Contents

List of Figures

List of Tables

Chapter

General Introduction

1. Research context
2. Objectives of the thesis
3. Thesis outline

Research context

In recent years, the rapid growth of mobile technologies, the IoT and interactive multimedia applications has profoundly transformed data processing, communication, and security requirements. The increasing volume of data exchanged, combined with the growing demand for real-time services, highlights the limitations of traditional cloud architectures. Although efficient in terms of centralized storage and computing power, these architectures suffer from inherent latency, network congestion, and a reliance on remote data centers. These constraints are particularly critical in contexts where responsiveness, security, and service availability are vital, such as in video surveillance systems, live streaming services, connected healthcare devices, or autonomous vehicles. In this context, the Mobile Edge Computing (MEC) paradigm is emerging as a relevant solution, bringing processing capabilities closer to end users, thus significantly reducing latency while relieving network congestion.

However, while MEC offers a significant performance advantage, it also raises new technological challenges. Available resources at the network edge are limited and heterogeneous, making it difficult to manage complex tasks (particularly security-related) efficiently. Furthermore, the massive deployment of connected devices multiplies attack surfaces and exposes networks to increased risks of intrusion, tampering, and loss of sensitive data. It is in this context that the integration of innovative technologies such as Software-Defined Networking (SDN) for centralized traffic management, Network Functions Virtualization (NFV) for dynamic resource orchestration, and, above all, Blockchain for secure exchanges, is attracting growing interest. When intelligently combined, these technologies enable the design of more flexible and secure service architectures that are adapted to dynamic environments. This is precisely the issue that our research work addresses, which lies at the intersection of these approaches, with the aim of proposing secure, efficient and scalable solutions for the offloading, management and storage of critical data in MEC environments.

Objectives of the thesis

The rapid evolution of mobile technologies, wireless networks, and multimedia services has led to an explosion in the volume of data exchanged, particularly in connected environments such as small cell networks, IoT devices, smart vehicles, and video surveillance systems. In this context, the MEC paradigm is emerging as a promising alternative to traditional cloud architectures, bringing computing and storage resources closer to end users. This approach reduces latency, relieves congestion in the core network, and optimizes QoS. However, despite these advantages, MEC introduces new vulnerabilities due to the decentralized and heterogeneous nature of its resources. Secure traffic management, reliable storage of sensitive data (particularly surveillance videos), and efficient execution of processing tasks are becoming major challenges. Traditional security solutions, often centralized and resource-intensive, do not scale well to distributed and constrained MEC environments.

Furthermore, the need for secure offloading of mobile data across multiple and complementary networks (small cells, Wi-Fi, D2D) is becoming increasingly urgent. Trust in connected devices, confidentiality of transferred data, and storage reliability must be guaranteed, while maintaining low network load and acceptable performance for sensitive services such as streaming or video surveillance. Moreover, the intelligent management of computing tasks (e.g., authentication, encryption, or integrity verification) in

a distributed environment with limited resources remains underexplored. This is the context in which this thesis aims to combine the advantages of SDN, Blockchain, and NFV technologies to propose a secure, distributed, and optimized MEC architecture. The objective is to design a solution that ensures secure multipath offloading, efficient execution of security tasks, and decentralized storage of sensitive data, while minimizing network overhead.

The objectives of this thesis are as follows :

- Propose a hybrid MEC architecture, based on SDN, Blockchain, and NFV technologies, capable of efficiently managing security tasks while maintaining high QoS.
- Develop a trust mechanism for IoT devices in a mobile environment to secure access to edge services.
- Design a secure, multipath offloading method on heterogeneous networks (small cells, Wi-Fi, D2D), leveraging distributed peer-to-peer SDN controllers.
- Implement a decentralized surveillance video storage system based on IPFS, with integrity validation ensured by blockchain (notably SegWit).
- Integrate smart contracts to automate access rights management and ensure the traceability of data access or modifications.
- Provide intelligent optimization of security task management (encryption, hashing, authentication) using network functions virtualization (NFV) to dynamically distribute loads between edge computing nodes.

The first contribution of this thesis is the proposal of a secure mobile data offloading architecture in small cell networks, leveraging SDN and Blockchain technologies. This architecture introduces a distributed control system based on SDN, allowing flexible and efficient management of data routing on multiple and complementary paths (Wi-Fi, D2D, small cells). Thanks to the blockchain, a trust and traceability mechanism is integrated to authenticate terminals, protect sensitive data transfers, and avoid attacks such as flow hijacking or tampering. This system also guarantees the consistency of security policies across the entire edge network, relying on a peer-to-peer logic between SDN controllers. The approach aims to maintain a high quality of service, while strengthening the confidentiality and integrity of offloaded data in heterogeneous and dynamic environments.

The second contribution focuses on the intelligent and distributed management of security tasks in the MEC environment, particularly for real-time streaming applications. By exploiting NFV, the thesis proposes a model for dynamic allocation of security-related computing tasks (encryption, hashing, authentication) between edge computing nodes. This model optimizes the use of limited MEC node resources, distributing loads in a balanced manner, and minimizing the impact on latency or quality of service. By integrating resilience and adaptability in the management of these functions, the solution reduces points of failure and allows automatic adaptation to traffic variations or specific user needs. This contribution demonstrates that distributed mechanisms, when intelligently orchestrated, can offer both performance and security in constrained MEC systems.

Finally, the third contribution of this thesis concerns the securing of video surveillance data storage in distributed MEC environments. For this, an innovative solution is proposed, combining distributed storage on IPFS (InterPlanetary File System) with integrity verification mechanisms via Blockchain, in particular through the SegWit model. Smart contracts are implemented to automate access management to stored videos, guarantee their immutability, and ensure a high level of traceability and transparency. This solution also promotes the scalability of the system, allowing horizontal growth in the number of nodes without compromising data security or consistency. The approach demonstrates that the combination of MEC, Blockchain and distributed storage technologies can effectively meet the growing requirements of confidentiality, traceability and performance in intelligent surveillance systems.

Thesis outline

The thesis is structured into four chapters that organize all the work carried out during our research. These chapters follow a logical progression, from the analysis of fundamental concepts to the proposal and evaluation of concrete solutions. The four chapters are detailed as follows.

First, in Chapter 1, we undertook an in-depth analysis of MEC environments, focusing particularly on QoS requirements and security issues. This chapter constitutes the state-of-the-art of the thesis, in which we highlight the limitations of traditional infrastructures in meeting the needs of modern, performance-intensive applications and identify the security vulnerabilities of MEC architectures, particularly with regard to resource management, authentication, and data confidentiality.

Chapters 2, 3, and 4 present our main contributions. Chapter 2 is dedicated to securing data offloading in small cell networks. In Chapter 3, we discuss the intelligent optimization of computing task management in MEC environments. Chapter 4 focuses on improving network performance and security in video surveillance data storage.

Chapter 1

Analysis of QoS and security issues in Mobile Edge Computing environments

Introduction

The emergence of new mobile applications, such as augmented reality, intelligent video surveillance, connected vehicles, and smart objects, has profoundly transformed the requirements of current networks. These applications require very low latency, high bandwidth, increased reliability, and enhanced security. Faced with these requirements, traditional cloud computing architectures are quickly showing their limitations, particularly due to the distance between end users and centralized data centers. It is in this context that MEC is emerging as a promising paradigm, bringing processing and storage capabilities closer to users, i.e., to the network edge.

MEC helps relieve congestion in the core network and reduce latency by processing data locally, thereby significantly improving QoS. However, this decentralization also introduces new challenges, particularly in terms of resource management, mobility, interoperability, and, above all, security. MEC environments, due to their distributed and heterogeneous nature, are particularly exposed to various threats, such as man-in-the-middle attacks, data tampering, and confidentiality breaches. This makes the integration of robust, dynamic, and scalable security mechanisms essential.

With this in mind, several related technologies are being leveraged to enhance the performance and security of MEC environments. Among them, SDN (Software Defined Networking) enables centralized and dynamic network management through a separation between the control plane and the data plane. NFV (Network Function Virtualization) facilitates the agile deployment of virtualized network functions on MEC nodes. Blockchain, meanwhile, introduces a distributed trust mechanism, guaranteeing the integrity of data and transactions without a central authority. Finally, IPFS (InterPlanetary File System) offers a distributed storage solution, ideal for scenarios where data resilience and verifiability are essential.

However, despite the growing integration of these technologies, current solutions still face several limitations. The lack of coordination between the various network components, the computational overhead of MEC nodes, the absence of security protocols adapted to the limited resources of IoT devices, and the challenges related to interoperability between platforms remain major obstacles. These shortcomings highlight the need to develop intelligent and secure approaches capable of dynamically adapting to changing execution contexts.

This chapter therefore provides a detailed state-of-the-art review of MEC and associated technologies, with a focus on QoS and security issues. It aims to identify the main vulnerabilities of current systems and lay the conceptual foundations that motivate the work presented in the following chapters.

1.1 Mobile Edge Computing: Principles and Challenges

Over the past decade, cloud computing has emerged as a dominant model in the IT world, based on the centralization of computing, storage, and networking resources in large-scale data centers. This model allows users, often equipped with low-capacity devices, to access virtually unlimited resources via the Internet, thus promoting the development of dynamic and scalable services [1]. The rise of this technology has been a major catalyst for many digital businesses: Amazon, for example, derives a substantial portion of its profits from its cloud infrastructure [2], while services like Dropbox have been able to grow rapidly thanks to the agility and flexibility offered by the cloud.

However, this centralization has certain limitations, particularly in terms of latency, core network overload, and mobility. In this context, a new decentralized approach has emerged: Mobile Edge Computing (as shown in the figure 1.1). This technology aims to bring processing and storage capabilities closer to users, by placing them as close as possible to the data sources, i.e., at the edge of the network [3]. With the explosion in the number of connected devices and the increase in their computing power, MEC appears as a promising solution for locally executing critical tasks in real time, while reducing the load on the core network. Thanks to this proximity, it becomes possible to significantly reduce transmission delays, making MEC particularly suitable for latency-sensitive applications [4], the Internet of Things (IoT) [5], or the user-centric Internet [6]. Today, this vision is attracting growing interest, both from academia and industry, by bringing together the disciplines of mobile computing and wireless communications around a common goal: to offer more responsive, distributed and contextualized computing.

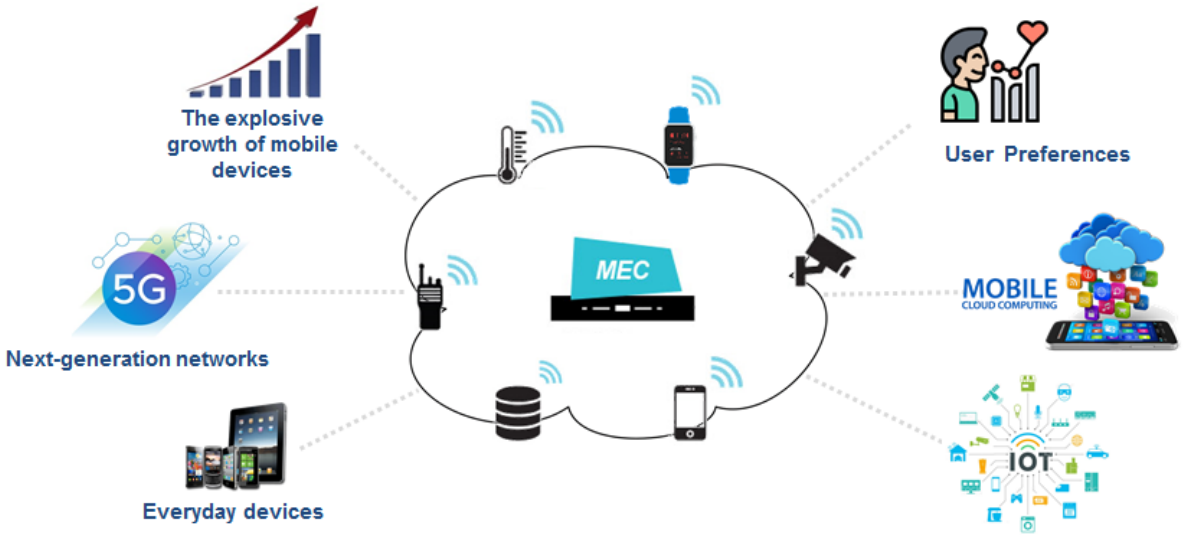


Figure 1.1: MEC as a technological change.

1.1.1 MEC Architecture

Mobile Edge Computing (MEC) architecture is part of a modern vision of mobile networks, driven by the rise of 5G and the proliferation of connected objects (IoT). Unlike the traditional approach of centralized cloud computing, MEC relies on distributed data processing at the network edge, closer to users and data-generating devices. This architecture aims to overcome the inherent limitations of the remote cloud, particularly in terms of latency, energy consumption, and bandwidth.

At the heart of the MEC architecture are edge computing nodes, typically deployed on network access entities such as base stations, Wi-Fi access points, or micro-data centers located at the edge of the network. These nodes, equipped with computing, storage, and connectivity resources, are capable of locally performing functions previously reserved for the cloud, such as real-time data analysis, image processing, or autonomous decision-making. This spatial proximity to users makes it possible to drastically reduce response time and guarantee low-latency services, essential for critical applications such as autonomous vehicles, augmented reality or telemedicine.

Figure 1.2 illustrates a generic three-layer hierarchical architecture of a distributed MEC environment for efficient data storage, processing, and control.

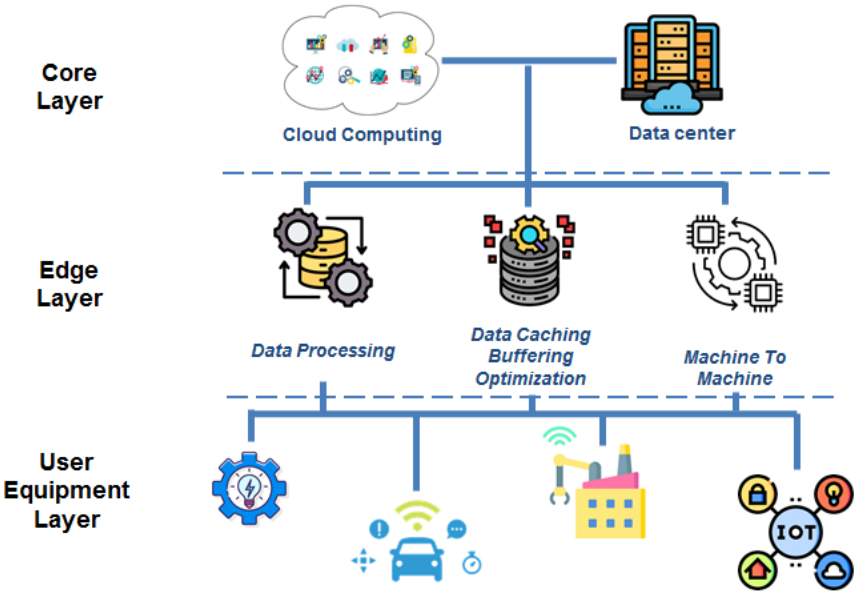


Figure 1.2: MEC architecture.

- The **User Equipment Layer (UEL)**, consisting of mobile devices, sensors, connected objects, and lightweight terminals. These devices continuously generate data but have limited resources, justifying the need to offload processing to MEC nodes.
- The **Edge Layer (MEC)**, where MEC servers are deployed. This layer ensures local data processing, execution of edge-aware applications, rapid decision-making, and

management of local resources (CPU, memory, energy). It can also integrate virtualized network functions (VNFs) to facilitate dynamic configuration and reconfiguration of services.

- The **Core Network/Cloud Layer** , which remains essential for massive storage, centralized services, offline analysis, and global learning. The MEC interacts with the cloud to delegate tasks that exceed its capacity or to synchronize the results of distributed processing.

This hybrid architecture enables flexible orchestration between the edge and the cloud, based on bandwidth constraints, application priority, and delay sensitivity. MEC also relies on complementary technologies such as Software-Defined Networking (SDN) and Network Function Virtualization (NFV), which facilitate abstraction [7], virtualization [8], and dynamic scheduling of the network and resources [9].

One of the distinctive aspects of MEC is the co-management of communication and computing resources. Decisions regarding service placement [10], task offloading [11], or function migration [12] must take into account both radio conditions (signal strength, interference, mobility) and computing constraints (CPU load, available memory, required latency). This need for joint optimization is a major area of research in MEC systems.

Furthermore, the heterogeneity of deployment scenarios makes MEC architecture particularly flexible. There are several variations: MEC with caching functions for content distribution [13], MEC for mobile environments with mobility management [14], ecological MEC integrating energy constraints [15], and privacy-oriented MEC [16], integrating privacy preservation mechanisms directly at the edge level.

Finally, numerous efforts are underway to standardize MEC architecture, particularly within the ETSI (European Telecommunications Standards Institute), which proposes a modular and interoperable reference architecture. This architecture defines the functional components, interfaces, APIs, and interaction procedures between layers. This standardization work aims to accelerate the industrial adoption of MEC by ensuring compatibility between equipment from different suppliers and operators

1.1.2 Data Offloading

With the explosion of mobile traffic generated by smartphones, connected objects, and data-intensive multimedia applications (HD video, online gaming, augmented reality), traditional mobile networks, and particularly 4G/5G infrastructures, are facing increasing saturation. Faced with this pressure, mobile operators are seeking effective mechanisms to optimize cellular infrastructures while maintaining the quality of service expected by end users. It is in this context that the concept of mobile data offloading has emerged as a crucial solution.

Mobile data offloading refers to a set of techniques that redirect a portion of cellular mobile traffic to alternative access networks, such as Wi-Fi, small cell networks, or Device-to-Device

(D2D) communications (show the figure 1.3). The objective is twofold: to relieve congestion on the primary mobile network and improve the user experience by reducing latency and increasing available bandwidth. This strategy is particularly relevant in dense environments (urban centers, public events), where demand often exceeds available radio capacity.

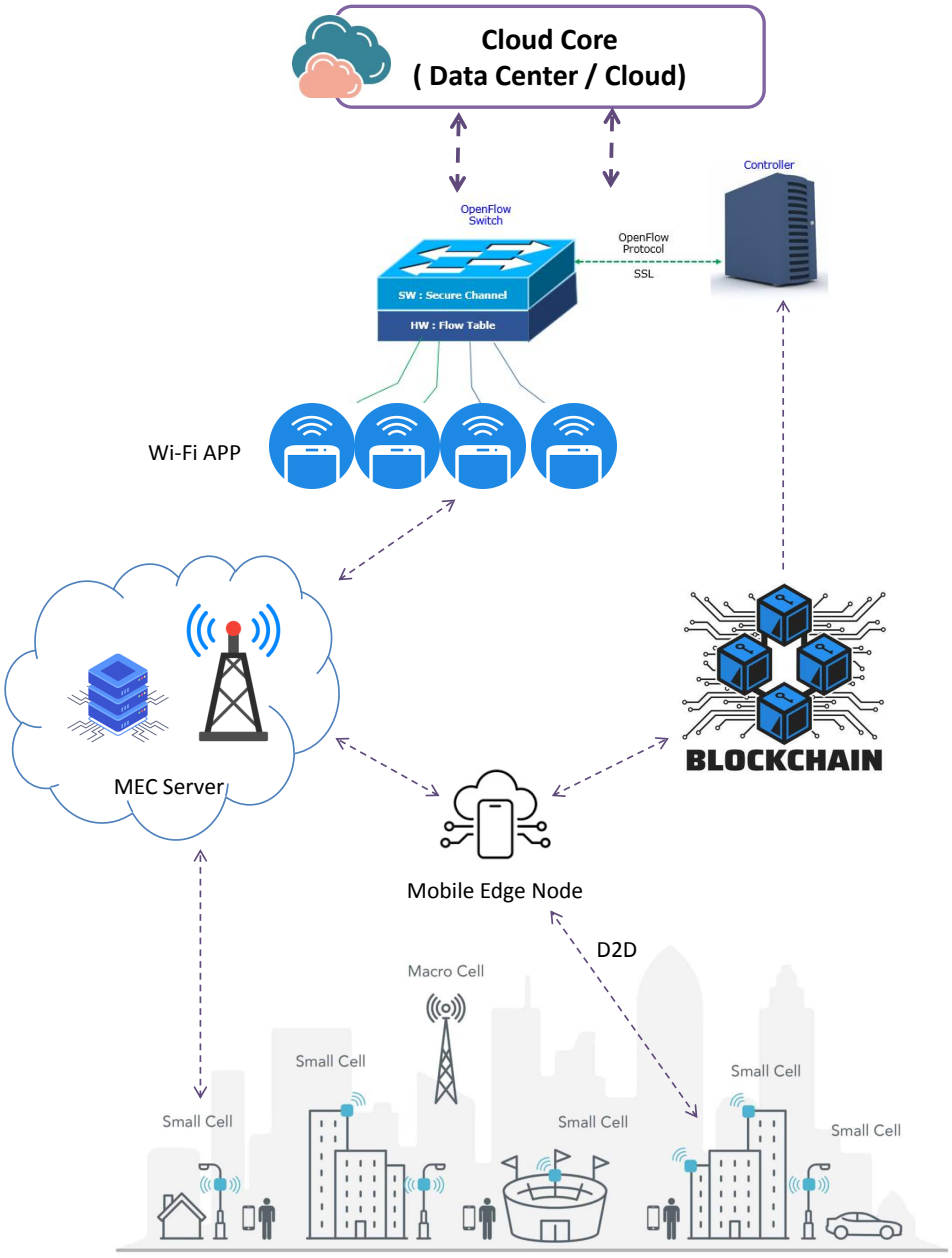


Figure 1.3: General architecture of mobile data offloading in a MEC environment. Data from mobile devices can be dynamically redirected to different complementary access networks (Small Cells, Wi-Fi, D2D) depending on load, latency, and security constraints. Control is provided by an SDN controller, and trust is maintained via a distributed blockchain.

Various offloading approaches have been proposed. The most common approach involves

automatically exploiting available Wi-Fi networks [17], using driven auto-association mechanisms [18]. In addition, more modern approach relies on the use of small cells [19] (pico, femto, or microcells), which provide high-capacity local coverage in high-traffic areas. Finally, emerging direct-to-device (D2D) communication technologies allow devices to exchange data without relying on central infrastructure [20], thus improving spectral efficiency and reducing latency.

However, implementing mobile data offloading poses several technical and security challenges. The main problem lies in the dynamic control and management of data flows between different network types [21]. Mobile users may be required to frequently change access points (handover), which requires intelligent mobility management [22] and seamless traffic orchestration [23] to avoid service interruptions or degradation. Furthermore, the use of public or third-party networks raises significant security issues: data interception, man-in-the-middle attacks, and access point spoofing.

To address these challenges, several complementary technologies are being leveraged. The SDN allows for centralized control of flow routing and dynamic adaptation of data paths based on network conditions [24]. Blockchain, for its part, can offer distributed trust, identity verification, and traceability mechanisms to guarantee the authenticity of access points and secure offloading sessions [25]. Furthermore, solutions based on artificial intelligence and contextual optimization can predict the best offloading times and points, while respecting QoS and security constraints [26].

Another critical aspect of mobile data offloading concerns traffic prioritization. Not all data is equal: some, such as video surveillance streams or medical data, require secure processing and reliable transmission with minimal latency. It then becomes necessary to implement traffic classification and differentiated processing policies capable of identifying sensitive data and choosing the most secure paths for its transmission [27]. This requires close integration between the network, application, and security layers.

Mobile data offloading cannot be separated from the evolution toward edge-aware architectures, such as MEC. By bringing data processing closer to the user, MEC not only reduces the load on the core network, but also allows critical processing (filtering, encryption, aggregation) to be performed locally before offloading [28]. This synergy between MEC and intelligent offloading constitutes a strategic lever for building mobile networks that are efficient, flexible, and resilient.

Mobile data offloading is also not limited to simple traffic redirection. It represents an adaptive network resource management mechanism [29], which requires complex coordination between access technologies, control mechanisms, and security policies. It is in this context that our work, presented in the next chapter, proposes an innovative secure data offloading architecture combining SDN, Blockchain, and complementary networks, in order to guarantee performance, security, and service continuity for mobile users in MEC environments.

1.1.3 MEC Advantages

As previously discussed, MEC represents advanced architectures for mobile networks, bringing computing, storage, and analytics resources closer to the end user. The decentralization of these architectures offers numerous advantages, both in terms of performance and quality of experience, thus meeting the growing demands of modern applications. These include:

- **Ultra-Low Latency**: One of the most significant benefits of MEC is the drastic reduction in latency. By processing data locally, at base stations, Wi-Fi access points, or edge micro-servers, MEC eliminates the need to transmit data to remote data centers, often hundreds of kilometers away. This proximity enables near-instantaneous response times, which is essential for time-sensitive applications such as telemedicine, autonomous driving systems, augmented reality, or interactive online gaming. These improvements in responsiveness translate directly into better QoE for the end user.

- **Optimizing bandwidth usage**: Another key advantage of MEC lies in optimizing bandwidth usage. By enabling local data processing and filtering, MEC prevents massive volumes of information (i.e., often raw or redundant) from being sent to the central cloud. This approach helps relieve uplink congestion, reduce the load on the core network, and improve the availability of network resources for other critical services. This is particularly relevant in dense urban environments, smart cities, or vehicular networks, where efficient bandwidth management is a key factor.

- **Energy efficiency**: MEC also contributes to improved energy efficiency of mobile devices. By limiting long-distance transmissions and promoting processing close to the device, MEC reduces the demand on the devices' radio modules, resulting in lower energy consumption. This optimization not only extends battery life but also improves the energy sustainability of the entire mobile ecosystem, which is particularly important in the current context of resource optimization and reducing the digital carbon footprint.

- **Service Personalization**: In addition, MEC facilitates the personalization of services based on the local context. Thanks to their proximity to the user, edge systems can leverage contextual information (location, network environment, usage history) to dynamically adapt application behavior. This paves the way for smarter, more responsive services that are better adapted to individual user needs, while enhancing security and privacy through localized processing of sensitive data.

1.1.4 MEC Limitations

Despite its many advantages, MEC also poses several technical, organizational, and security challenges that limit its large-scale deployment. These challenges must be identified and

overcome to ensure sustainable, reliable, and interoperable adoption of this technology in heterogeneous and highly constrained environments.

- **Distributed resource management**: Unlike a centralized cloud, where resources are massive and shared, MEC nodes have limited processing, memory, and storage capacities. It is therefore crucial to design intelligent mechanisms for service placement, orchestration, and migration to ensure optimal task distribution between the different network layers (edge and cloud). Dynamically managing these resources in constantly changing environments (user mobility, load variations, network outages) represents a major combinatorial optimization challenge.

- **Data security and confidentiality**: By bringing processing closer to end users, MEC increases exposure to the risks of intrusion, manipulation, or theft of sensitive data. Edge nodes, often deployed in areas without physical protection, can be prime targets for attacks. Furthermore, traditional security mechanisms designed for the cloud (such as centralized firewalls or mass intrusion detection systems) are unsuitable for the distributed and constrained environments of MECs. It is therefore necessary to implement lightweight, distributed, and autonomous solutions capable of ensuring authentication, encryption, traceability, and access management, while respecting latency and energy consumption constraints.

- **User mobility**: This limitation also represents a specific challenge. In highly mobile environments, such as vehicular networks or intelligent transportation, connections between terminals and edge nodes are often unstable. This requires the implementation of transparent handover strategies and seamless migration of tasks to new anchor points, without service interruption or performance loss. This requires inter-node mobility management protocols capable of anticipating movements and optimizing network resources in real time.

- **Interoperability**: Interoperable MEC solutions are a major challenge in a context where equipment is supplied by multiple manufacturers and operators each have their own management policies. The lack of universally adopted standards can hamper the integration of MECs into existing network architectures. Although standardization efforts are underway, notably through ETSI MEC, the seamless integration of MECs with technologies such as 5G, SDN, blockchain, and NFV remains a complex architectural challenge.

- **Economic limitation**: The monetization of MECs and the definition of viable business models for operators and service providers remain uncertain. The investment required to deploy edge nodes on a large scale is significant, and pricing, resource sharing, and service guarantee mechanisms remain poorly defined.

Although MEC offers promising prospects for the next generation of smart grids, it still requires significant efforts in terms of research, standardization, security, mobility management, and deployment models. These limitations constitute avenues of research that motivate the contributions proposed in this thesis.

1.1.5 Challenges and perspectives

The development of MEC marks a decisive step in the evolution of network architectures and distributed processing. However, beyond the identified limitations, MEC faces structural challenges that must be addressed to ensure its large-scale deployment, as well as promising research opportunities to explore in the coming years.

- **Intelligent and adaptive resource management**: MEC operates in a distributed, heterogeneous, and often dynamic environment, where processing and bandwidth demand varies continuously. It is becoming essential to design autonomous orchestration algorithms capable of taking into account network load, application priorities, radio conditions, and user mobility in real time. Integrating AI and machine learning techniques to predict demand and anticipate service migration represents a particularly promising avenue of research.

- **Distributed security of MEC environments**: As network and application functions become more decentralized, the attack surface expands. New embedded security mechanisms capable of operating on resource-limited edge nodes must be developed. Technologies such as blockchain, smart contracts, and distributed trust protocols are increasingly being considered to enhance the transparency, traceability, and resilience of edge processing. However, these mechanisms must be adapted to the strict constraints of MEC: low latency, low power consumption, interoperability, and ease of integration.

- **Standardizing MEC**: Despite the efforts of ETSI MEC and industrial consortia, current solutions are still immature enough to ensure complete interoperability between the various layers and entities of the system (cloud, edge, terminal, operators, service providers). It is therefore essential to continue standardization efforts while developing modular, open, and extensible architectures capable of adapting to rapid changes in uses and technologies (5G, IoT, cloud-native, embedded AI).

The MEC opens up numerous opportunities for innovation in critical application areas: the evolution of smart grids, particularly in the era of 5G, the Internet of Things (IoT), and artificial intelligence (AI). Its gradual adoption by manufacturers and researchers opens up a wide range of technological and scientific opportunities for the years to come. MEC is not simply an extension of the cloud to the edge, but a paradigm shift that reconfigures the roles, models, and possibilities of digital networks. The coming years will undoubtedly see the emergence of

autonomous, secure, intelligent, and energy-efficient MEC platforms capable of supporting the growing demands of tomorrow's applications.

One of the major opportunities for MEC is its close integration with 5G and future 6G networks. Thanks to base station densification and network slicing, MEC can provide personalized, ultra-low-latency services for critical applications such as remote surgery, autonomous vehicles, and tactical networks. MEC could thus become a pillar of ultra-reliable, ultra-low-latency (URLLC) networks, enabling decision-making directly at the edge.

Furthermore, MEC plays a key role in the emergence of intelligent autonomous networks. By locally deploying analysis, prediction, and decision-making capabilities, MEC enables the implementation of autonomous and adaptive architectures capable of self-configuring services, dynamically optimizing resources, and interacting in real time with their environment. The combined use of machine learning and edge AI will make it possible to develop systems capable of anticipating user needs, detecting anomalies in real time, and coordinating actions in a decentralized manner.

Another key aspect of MEC concerns digital sovereignty and the protection of personal data. By processing data locally, as close as possible to its source, MEC significantly reduces the risks associated with transmission to centralized clouds. This approach is particularly attractive in sensitive sectors (healthcare, defense, finance), where confidentiality and local control of data are crucial issues. Combined with technologies such as blockchain, MEC could constitute an essential building block for distributed systems for managing trust, data authenticity, and governance.

MEC also paves the way for innovative business models, particularly in the context of Edge-as-a-Service (EaaS). Operators, businesses, or local authorities will be able to offer processing, storage, or security services at the edge, intended for third-party users or vertical applications (industry, agriculture, energy, smart cities). This distributed vision of the digital economy still requires advances in standardization, interoperability, and automatic contracting (via smart contracts).

In addition, the MEC opens up numerous opportunities for innovation in critical application areas. Connected vehicles (V2X), digital health, drone networks, extended reality (XR) and even smart factories (Industry 4.0) are all contexts where ultra-localized data processing is becoming essential. Ultimately, MEC could become a fundamental basis for autonomous networks, capable of making decisions locally without relying on a remote center. Future work on the MEC will also need to address issues of energy sustainability and environmental impact. Optimizing the energy consumption of edge nodes, pooling physical resources, and integrating renewable energy into peripheral deployments are all projects that will determine the eco-responsible nature of the networks of the future.

1.2 Associated technologies

Effective MEC deployment relies on the integration of several cutting-edge technologies that contribute to optimizing resource management, network flexibility, data security, and service decentralization. Among these technologies, four pillars are particularly important: Software Defined Networking (SDN), Network Functions Virtualization (NFV), Blockchain, and the InterPlanetary File System (IPFS). This section presents the role, principles, and contributions of each of these technologies in an MEC environment.

1.2.1 Software Defined Networking (SDN)

SDN standardization activities were initiated and supported primarily by the Open Networking Foundation (ONF) [30], a non-profit organization bringing together industry and academic stakeholders. The ONF played a key role in defining the fundamental concepts of SDN and in developing the OpenFlow protocol, widely recognized as the first standard interface for communication between the control plane and the data plane. This protocol provided a starting point for the implementation of interoperable SDN architectures, facilitating the adoption of this paradigm worldwide [31, 32].

At the same time, other standards bodies contributed to the evolution of the field. The Internet Engineering Task Force (IETF) worked on complementary protocols for network management and automation, such as NETCONF and YANG, used for device configuration. For its part, the ETSI (European Telecommunications Standards Institute) has integrated SDN principles into its work on Network Functions Virtualization (NFV), seeking to establish synergies between network programmability and virtualization. These joint efforts aim to ensure multi-vendor interoperability, reduce dependence on proprietary solutions and accelerate the transition to flexible and programmable networks [33, 34].

The ONF proposes this most adopted definition: "*The SDN constitutes an emerging network architecture in which the control function is separated from the transfer plan and made directly programmable [35]*". In other words, the SDN represents a conceptual break in the way of designing, deploying and administering network infrastructure. Unlike conventional networks, it is based on a dissociation of the control plan and the data plan, which allows centralized, flexible and programmable management of the overall behavior of the network. This management is ensured by an SDN controller, responsible for supervising and coordinating all the equipment (switches, routers, etc.).

SDN architecture:

An SDN as a network architecture is defined with three layers: the infrastructure layer, the control and application layer. The figure 1.4 presents the logical view of the SDN architecture. In this model, the infrastructure layer transmits the control information to the SDN management

software located in the control layer, using dedicated interfaces. This interaction allows us to build an abstract view of the network while ensuring the update of the configuration and state of the underlying infrastructure. On this basis, network services are generated by operating the data available at the SDN controller. Business applications then access the network configuration and information related to infrastructure via API interfaces, guaranteeing flexible and scheduled interaction with network services. Unlike traditional networks, where access to information is generally limited, manual, and specific to each device, the SDN architecture promotes the exchange of information in real time, allowing for automated and intelligent processing due to the integration of algorithms and adapted programs.

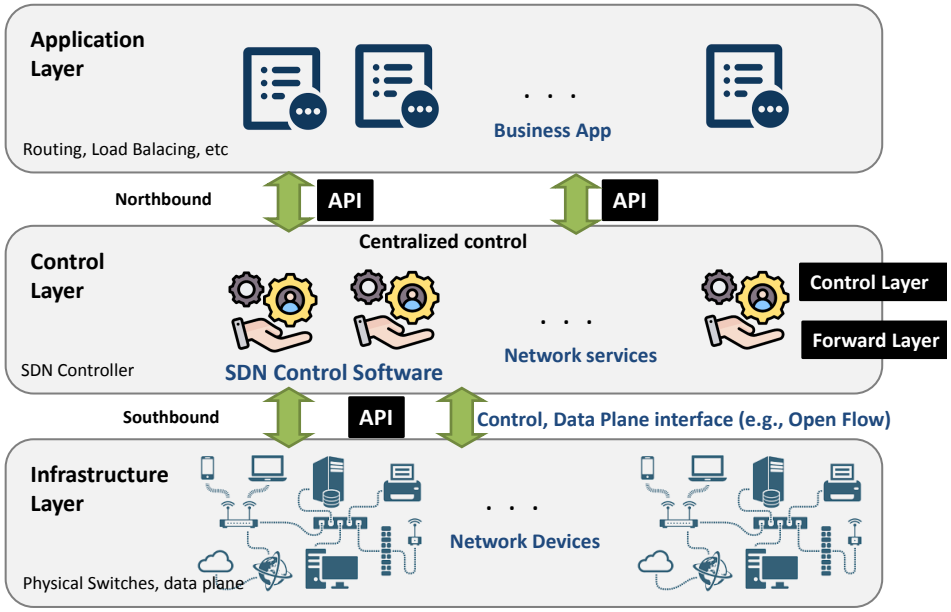


Figure 1.4: Software-Defined Network

- **Infrastructure Layer :** the SDN introduces a radical transformation in the way the networks are designed and administered. Unlike traditional infrastructure, where communication equipment has both transfer functions and decision -making mechanisms, the NTC operates a functional dissociation between these two plans. Physical devices, such as routers, switches or middleboxes, are assigned a minimalist role, limited to the simple processing and routing of packets, while the intelligence of the network is fully centralized in a software controller. This centralization gives a global and coherent view of the system, paving the way for more flexible and adaptive management strategies.

- **Control Layer:** In an SDN architecture, the control layer occupies a central position as a strategic component of the network. It is not limited to simple routing instructions, but constitutes the decision -making space where global operating rules are developed. This

layer is generally embodied by the SDN controller and the software applications that use it. Its role is to translate the needs expressed by applications or administrators in traffic management policies, which are then transmitted to the data plan equipment. In this sense, it acts as the brain of the network, with an abstract and consolidated view of the underlying infrastructure. Thanks to this global vision, it can optimize the use of resources, guarantee quality of service and ensure dynamic adaptation to load variations or unforeseen events.

- **Application Layer:** (or Management Plane (MP)) is the set of software applications that leverage the functionality offered by the northbound interface to define and enforce network operating rules. This plane is the level where administration and control logic is expressed, through various services such as routing, security (firewall), load balancing, and performance monitoring. In concrete terms, management applications translate global policies into a set of operational instructions, which are transmitted to data plane devices to configure and direct their behavior.

Another pillar of this architecture lies in the use of open and standardized interfaces, including OpenFlow is an emblematic example. These interfaces ensure not only the direct programmability of heterogeneous equipment, but they also promote interoperability between technologies and manufacturers, thus breaking the limitations imposed by proprietary solutions. This dynamic configuration capacity transforms the network into a programmable platform on which applications can be deployed to meet evolutionary needs, whether it is optimizing performance, strengthening security or supporting new services. We distinguish:

- **Southbound Interface (SBI):** refers to the set of mechanisms that enable communication between the control plane and the data plane. It relies on standardized APIs and protocols that specify both the instructions sent to the forwarding devices and how they return their status to the controller. The goal is to establish a clear and structured interaction, ensuring that the controller can efficiently control the physical devices of the network infrastructure.

- **Northbound Interface (NBI):** constitutes the point of contact between the SDN controller and network applications. It provides developers with a unified and abstract framework for expressing high-level policies or requirements, without having to worry about the details of the hardware implementation. In this way, the NBI simplifies the development of innovative applications by masking the complexity of the low-level interactions provided by the SBI, and fosters the emergence of programmable and scalable network services.

OpenFlow Protocol:

The OpenFlow protocol is widely recognized as the first communication standard in the SDN paradigm. It defines a standardized interface between the control plane and the data plane,

allowing an SDN controller to directly control the behavior of switches and routers. Unlike traditional approaches where each network device integrates its own control mechanism, OpenFlow moves this logic to a centralized software entity, promoting innovation, programmability, and simplifying network management. According to the ONF, OpenFlow plays a key role as the founding protocol of SDN. It provides a set of standardized instructions allowing controllers to install, update, and delete packet processing rules in data plane devices. Thanks to this abstraction, OpenFlow allows the creation of dynamic, interoperable network services independent of proprietary solutions, thus constituting a pillar of modern SDN architecture.

The OpenFlow architecture is designed to achieve the separation of the control plane and the data plane in SDN. It is based on two fundamental elements: the SDN controller and OpenFlow switches, connected by a secure communication channel using the OpenFlow protocol.

SDN Controller (Control Plane): The controller is the central decision-making entity in the network. It maintains a global and abstract view of the infrastructure, allowing it to define policies for routing, security, or quality of service management. These policies are then translated into flow rules and transmitted to the switches via the OpenFlow interface. The controller therefore acts as the brain of the network, dynamically coordinating and adapting the overall system behavior. Several controllers have been developed in different contexts: some for education and research (such as POX and Ryu), others for enterprise production environments (Floodlight, OpenDaylight), and finally some for operators and large infrastructures (ONOS, Stratum). The table 1.1 presents a summary comparison between some popular SDN controllers.

As noted, SDN controllers differ in terms of development language, scalability, and use cases. Lightweight controllers like POX and Ryu are suited to academic and research environments, thanks to their simplicity and ease of programming. On the other hand, Floodlight and OpenDaylight offer better integration for production environments, with broader network protocol support. Finally, ONOS and Stratum target large-scale mission-critical environments, such as telecom, 5G networks, and IoT, where resilience and programmability are essential.

OpenFlow Switches (Data Plane):

Switches are responsible for the actual routing of packets. They do not make any autonomous decisions, but execute instructions received from the controller. Each switch contains one or more flow tables that define how packets should be processed. When a packet arrives, it is compared against existing rules; if no match is found, a query is sent to the controller to obtain a new, suitable rule. An OpenFlow switch is composed of three main components (1.5):

1. *Flow Table:* Contains the routing rules defined by the controller. Each rule specifies conditions (e.g., source IP address, MAC address, TCP port) and an action (forward, redirect, modify the header, or drop the packet).

2. *Secure Channel:* Establishes encrypted communication with the controller via the OpenFlow protocol (usually TCP/TLS).

Table 1.1: Comparison of popular SDN controllers

Controller	**Main language**	**Scalability**	**OpenFlow support**	**Use cases**
NOX	C++, Python	Low	Yes (1.0–1.3)	Early research, historical use
POX	Python	Low	Yes (1.0–1.3)	Education, simulation (Mininet)
Floodlight	Java	Medium	Yes (1.0–1.3)	Enterprise production, REST API
Ryu	Python	Medium	Yes (1.0–1.5)	Research, rapid prototyping
OpenDaylight (ODL)	Java	High	Yes + other protocols (NETCONF, BGP, etc.)	Enterprise and industrial networks
ONOS	Java	Very high	Yes + P4, NETCONF, gNMI	Telecom operators, 5G, IoT
Stratum	Go, C++	Very high	Replaces OpenFlow (P4Runtime, gNMI)	Next-generation programmable networks

3. *OpenFlow Protocol:* Enables the exchange of instructions, statistics, and events between the controller and the switch.

Thus, the OpenFlow switch executes only the received rules, without local control logic, which centralizes network intelligence at the controller level. For instance, *Open vSwitch (OVS)* is one of the most widely used OpenFlow switches, implemented in software. It is widely used in virtualization environments (KVM, Xen) and cloud solutions (OpenStack). *HP 5400zl* and *HP 8200zl* are physical switch ranges that support OpenFlow for integration into hybrid SDN networks. *NEC ProgrammableFlow* is a hardware solution based on OpenFlow, often used in carrier networks. *Pica8 Switches* are OpenFlow-compatible physical devices, frequently deployed in data centers.

Based on this architecture, OpenFlow offers fine-grained and dynamic network programmability, making it possible to implement advanced applications such as adaptive routing, intrusion detection, load balancing, or QoS management. This approach differs radically from traditional networks, where each device has its own control plane, often closed and proprietary.

MEC-SDN

Integrating SDN with MEC represents a promising approach to improve the performance of next-generation networks, especially in highly dynamic environments such as the Internet of

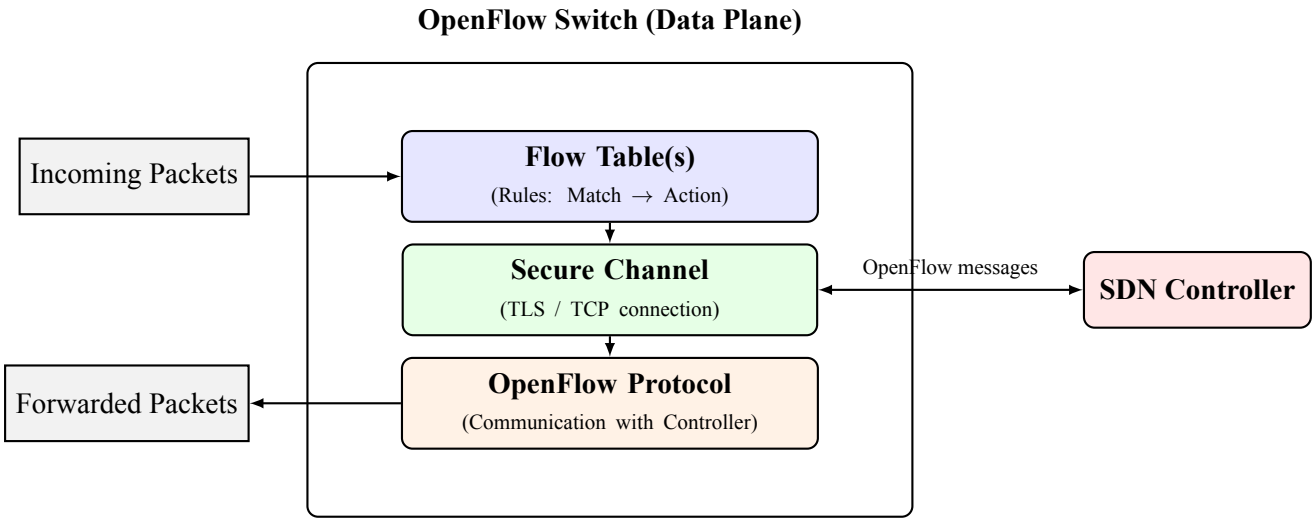


Figure 1.5: Internal structure of an OpenFlow switch (Data Plane).

Things (IoT), 5G networks, and beyond. The key benefit of this combination is that MEC brings computing and storage resources closer to the end user, while SDN provides network programmability and agility to quickly redirect flows to the nearest MEC server. For example, a moving connected vehicle can switch its connection from one MEC station to another without service disruption, thanks to the dynamic reconfiguration of routing rules managed by the SDN controller. In this architecture, the network is organized into three main layers:

1. **Data Plane:** Consisting of communication devices (base stations, routers, OpenFlow switches) located both in the network core and at the edge near the user. These devices forward packets according to the rules received from the control plane.

2. **Control Plane (SDN Controller):** Network intelligence is centralized in one or more SDN controllers. These dynamically manage connectivity between users, MEC servers, and the network core. They leverage the OpenFlow protocol (or other open interfaces) to schedule transmission devices and ensure fine-grained flow orchestration.

3. **Application and Orchestration Layer:** This layer encompasses MEC services hosted near users (e.g., augmented reality, video streaming, connected vehicles, e-health). Using northbound APIs (northbound interfaces), applications interact with the SDN controller to request optimized network resources based on load, latency, and expected QoS.

In the context of MEC, SDN plays a key role by enabling flexible orchestration of data flows between edge nodes and the cloud. It facilitates the dynamic programming of routing rules, traffic adaptation based on network conditions, and the implementation of QoS policies tailored to the needs of latency-sensitive applications. Thanks to this programmable architecture, operators can respond quickly to load variations, network failures, or topology changes induced by user mobility.

One of the most significant contributions of SDN to MEC systems is its ability to centralize the management of distributed resources while providing a holistic view of the network. This helps optimize routing decisions, reduce latency, and improve energy efficiency. Furthermore, SDN facilitates the integration of advanced security features such as dynamic filtering, anomaly detection, and malicious traffic isolation, which is essential in distributed and open environments like MEC.

SDN provides a natural bridge to other key MEC technologies, such as NFV, by enabling dynamic instantiation of virtualized network services, or blockchain, by facilitating the secure distribution of control policies. In short, SDN provides the agility, flexibility, and intelligence required to fully leverage the MEC paradigm.

1.2.2 Network Functions Virtualization (NFV)

Network Function Virtualization (NFV) represents an important advance in the way modern networks are designed and deployed. Its central principle is to replace specialized hardware (such as firewalls, routers, load balancers, and monitoring systems) with software functions running on generic infrastructures such as standard servers, cloud environments, or virtual machines. This separation of network functions from proprietary hardware significantly reduces the reliance on expensive and inflexible solutions based on dedicated hardware.

NFV's objective is twofold: first, to reduce capital expenditures (CAPEX) and operational costs (OPEX) associated with managing and renewing network equipment, and second, to provide greater flexibility and agility in service deployment. By virtualizing data plane functions, operators can dynamically deploy new services, quickly adjust resources based on load, and ensure greater scalability.

The need to define common foundations for NFV interoperability and performance led, as early as 2012, to a mobilization of major players in the sector. Several telecommunications operators and service providers published a founding white paper that led to rapid adoption of this approach. Following this, the European Telecommunications Standards Institute (ETSI) created the NFV Industry Specification Group (NFV ISG) [36], which continues to be the leading standardization initiative in this field today [37, 38]. The ETSI NFV ISG proposed a reference architectural framework, structured around three fundamental blocks: Virtualized Network Functions (VNFs), NFV Infrastructure (NFVI), and Management and Orchestration (MANO). These specifications established a common language between suppliers and operators, facilitating the interoperability and integration of heterogeneous solutions.

To ensure interoperability, performance, and widespread adoption of this technology, several standardization initiatives have emerged.

1. **ETSI NFV ISG (Industry Specification Group):** The most significant initiative was launched by the European Telecommunications Standards Institute (ETSI) in 2012, with the creation of the NFV ISG. This industry specification group defined:

- the NFV reference architecture, composed of three main blocks: Virtualized Network Functions (VNF), NFV Infrastructure (NFVI), and NFV Management and Orchestration (MANO).
- a set of specifications covering function virtualization, VNF lifecycle management, resource orchestration, and interoperability with cloud environments.

2. **OASIS, IETF et ONF:** The IETF (Internet Engineering Task Force) is working on the protocols and mechanisms needed to support virtualization, including service and description models (YANG, NETCONF). OASIS contributes with standards related to service management and orchestration (e.g., TOSCA for describing cloud topologies). The ONF, known for SDN, is also collaborating with ETSI to ensure compatibility between SDN and NFV.

3. **ETSI NFV Release Evolution:** Since its creation, the ETSI NFV ISG has published several ”Releases” (Release 1, Release 2, etc.) progressively integrating specifications for: resilience and high availability, management automation, integration with 5G Core and Edge Computing, compatibility with native cloud and containers (Kubernetes, Docker).

The NFV architecture, defined by ETSI, is based on a clear separation between network functions and the underlying hardware. Unlike traditional approaches where network functions (firewalls, NAT, DPI, EPC, etc.) were deployed on dedicated proprietary hardware, NFV introduces the virtualization of these functions by running them as software on standard servers, storage platforms, and cloud infrastructures.

The first key component is the Virtualized Network Function (VNF). A VNF corresponds to a network function implemented in software, for example, a virtual firewall or a virtual gateway. VNFs are instantiated, managed, and dynamically interconnected to meet traffic and service needs. They allow for great flexibility since they no longer depend on specific hardware.

The second fundamental building block is the NFV Infrastructure (NFVI). It is the set of hardware resources (servers, storage, network equipment) and the virtualization layer (hypervisors or containers) that enable the execution of VNFs. NFVI must offer high performance, isolation between instances, and dynamic adaptation capacity to support variable loads.

Finally, NFV Management and Orchestration (MANO) is the central control and coordination element. This entity combines three main functions:

- NFV Orchestrator (NFVO): Responsible for orchestrating network services and managing the VNF lifecycle.
- VNF Manager (VNFM): Responsible for instantiating, configuring, updating, and terminating VNFs.

- Virtualized Infrastructure Manager (VIM): Manages NFV resources, such as OpenStack or Kubernetes.

With this architecture, NFV provides operational agility and reduces capital expenditures (CAPEX) and operational costs (OPEX). For example, a mobile operator can deploy a service chain consisting of a virtual firewall, IDS, and load balancer, without the need to purchase specific hardware. This also allows resources to be quickly adapted to traffic spikes, such as during sporting events or service launches.

In the context of Mobile Edge Computing, NFV offers the ability to deploy specific functions at the network edge, tailored to local contexts, service types, and available hardware capabilities. For example, an edge node in a dense urban area might run a video prioritization service, while a node in a rural area might run a compression or preprocessing service. NFV also helps reduce infrastructure and maintenance costs by replacing rigid proprietary hardware with software solutions on generic servers. Scalability is also a major advantage: functions can be duplicated or migrated dynamically based on workload or user location. Another key advantage is resilience. If a virtualized service fails, it can be automatically restarted elsewhere in the infrastructure. This is particularly relevant in MEC architectures where fault tolerance is essential to ensure service continuity.

NFV integrates naturally with SDN, as the SDN controller can orchestrate flows to locally deployed virtualized functions. It is also compatible with automation and intelligent management systems (such as NFVO orchestrators) that enable global network monitoring.

1.2.3 Blockchain

Blockchain, as an emerging technology, is generating global interest to ensure interoperability, security, and widespread adoption. Given the diversity of use cases (finance, healthcare, logistics, IoT, energy, etc.), several standardization initiatives have been launched to define technical standards, regulatory frameworks, and best practices. The main objective of this work is to avoid technological fragmentation and foster an open and reliable ecosystem.

One of the main initiatives comes from ISO (International Organization for Standardization), which created the technical committee ISO/TC 307 in 2016 [39]. This committee focuses on standardizing terminology, reference architectures, digital identity, security, and governance in blockchain and Distributed Ledger Technology (DLT). The standards produced by this committee aim to facilitate the interoperability between different blockchain platforms and to encourage adoption by industry and governments. At the same time, the ITU-T (International Telecommunication Union – Telecommunication Standardization Sector) has established the Focus Group on Application of Distributed Ledger Technology (FG-DLT) [40]. This initiative explores the use of blockchain in telecommunications, particularly for trust management, cybersecurity, and service authentication. The ITU-T focuses on the integration of blockchain into next-generation networks and the IoT ecosystem. The ETSI (European Telecommunications

Standards Institute) is also contributing to this dynamic with its ISG PDL (Permissioned Distributed Ledger) group [41], which is developing specifications for permissioned blockchains. The goal is to meet the needs of network operators and European industries, particularly in terms of regulatory compliance and data management. In addition to these standards bodies, industry initiatives play a key role in de facto standardization. The Hyperledger Project (under the Linux Foundation) [42] provides an open-source framework for developing modular and interoperable enterprise blockchains. Similarly, the Enterprise Ethereum Alliance (EEA) [43]establishes standards for the use of Ethereum in an enterprise context, ensuring compatibility and robustness in industrial deployments. Finally, some government and regional initiatives aim to harmonize the use of blockchain. For example, the European Union is working on regulatory frameworks such as the European Blockchain Services Infrastructure (EBSI) [44], intended to provide a trusted infrastructure for digital public services. These efforts demonstrate that blockchain standardization is not limited to technical aspects, but also encompasses legal, ethical, and organizational dimensions.

Blockchain Architecture

A blockchain architecture is based on a distributed structure where data is organized in the form of cryptographically linked blocks. Each block contains a set of validated transactions, a timestamp, a cryptographic hash of the previous block, and a unique identifier. This organization ensures the integrity and traceability of information, since any attempt to modify a block would invalidate the entire chain. A blockchain can be broken down into several functional layers (as shown in the figure 1.6):

1. **The Network Layer:** This layer constitutes the foundation of the blockchain. It ensures: Peer-to-peer (P2P) communication between network nodes, propagation of transactions and blocks, synchronization and node discovery mechanisms. Without this layer, nodes would not be able to exchange information in a decentralized manner.

2. **The Data Layer:** This represents the fundamental structure of the blockchain; the blocks, which contain a header (hash of the previous block, timestamp, nonce, etc.) and a body (list of transactions). The blockchain is formed by the cryptographic chaining of successive blocks.

3. **The Consensus Layer:** This is the heart of the blockchain, responsible for validating and ordering transactions. It defines: the rules for all nodes to agree on a single state of the ledger, the protocols used, such as Proof of Work (PoW), Proof of Stake (PoS), or variants (PBFT, DPoS, etc.). This layer ensures that the system remains reliable and resilient to attacks, even in a distributed and untrusted environment.

4. **The Execution Layer (Smart Contract Layer):** This layer is responsible for the execution of validated transactions, the operation of smart contracts that automate certain

rules or applications (e.g., Ethereum), and the interaction between users and decentralized applications (dApps).

5. **The Application Layer:** This layer constitutes the interface visible to users. It includes decentralized applications (dApps) such as financial services (DeFi), NFTs, IoT, identity management, etc. In addition, it includes APIs and development tools for interacting with the blockchain, as well as business services that leverage the trust and transparency offered by the blockchain.

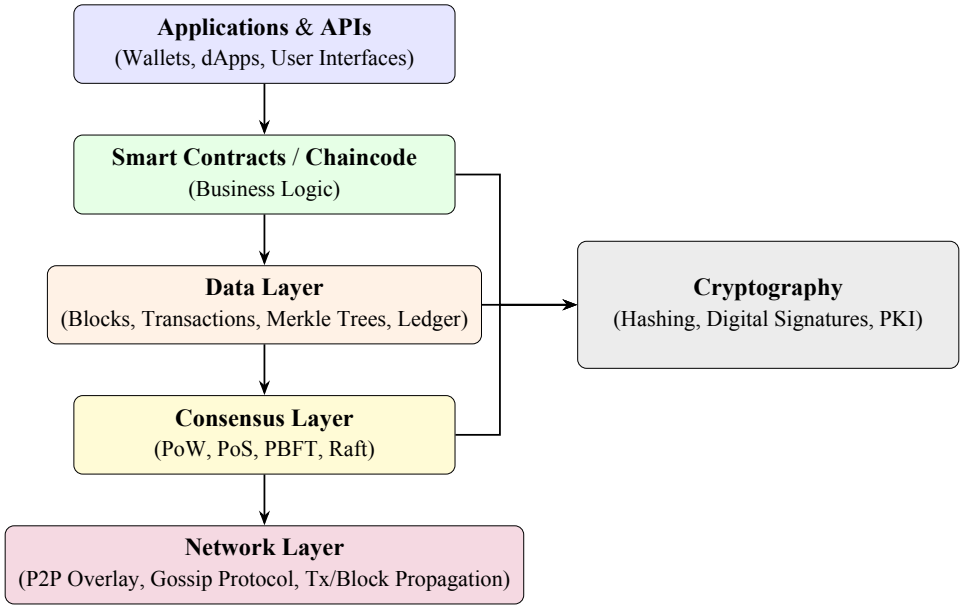


Figure 1.6: Compact layered architecture of a blockchain system.

A block is the fundamental unit of storage in a blockchain. Each block contains a set of validated transactions and essential information that ensures the integrity and continuity of the chain. The figure 1.7 shows the typical structure of a block which can be divided into two main parts: the block header and the block body.

1. **Block Header:** The header contains the metadata needed to identify and secure the block. Some of the most important fields include:

   - *Previous Block Hash:* Cryptographic reference that links the current block to the previous one, ensuring the continuity and immutability of the chain.
   - *Merkle Root:* Root of the Merkle tree, which summarizes all the transactions contained in the block. This cryptographic structure allows for quick and efficient verification of transaction integrity.
   - *Timestamp:* Indicates the approximate date and time the block was created.
   - *Nonce:* Value used in the consensus process, particularly in blockchains using Proof of Work, to validate the block hash.

- *Difficulty or Target:* Parameter indicating the difficulty of the mining process or the conditions required to validate the block.

2. **Block Body:** The block body contains the list of validated transactions recorded in the blockchain. Each transaction typically includes the sender's address, the recipient's address, the amount transferred, and the digital signature guaranteeing its authenticity. Depending on the blockchain, the block may also include additional data such as smart contracts, scripts, or metadata.

Block chaining is made possible by using the hash of the previous block stored in each header. Therefore, any modification to a transaction in a block would invalidate its hash and, consequently, break the chain's continuity. It is this mechanism that ensures the security, integrity, and immutability of the data recorded in the blockchain.

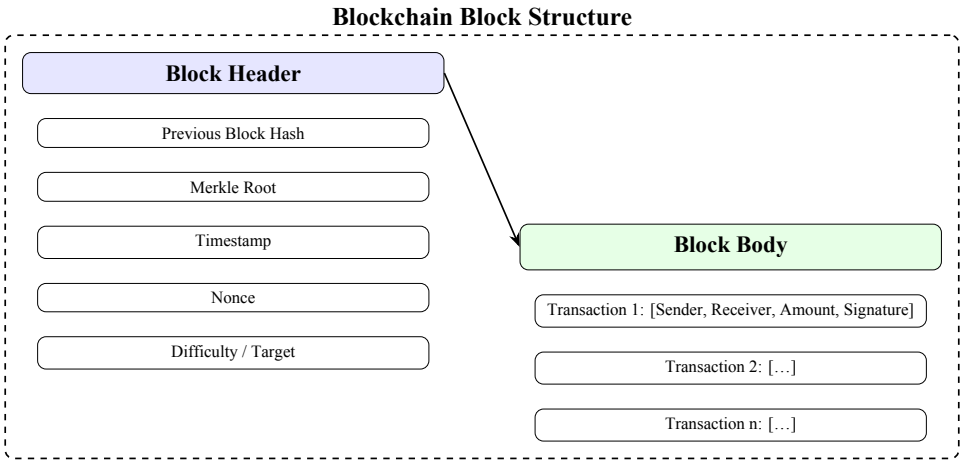


Figure 1.7: Structure of a blockchain block: header and body.

Finally, the blockchain architecture integrates security and cryptographic mechanisms. Digital signatures, based on asymmetric algorithms (such as ECDSA), guarantee the authenticity and non-repudiation of transactions. Hash functions ensure data integrity and prevent tampering. These combined elements give blockchain unique properties of decentralization, transparency, immutability, and censorship resistance.

By integrating blockchain into edge nodes, it is possible to decentralize the management of terminals' digital identities, store the fingerprints of sensitive files (logs, videos, personal data), or govern exchanges between IoT entities via smart contracts. These mechanisms are essential for environments with high autonomy or where multiple actors interact without mutual trust. Tamper resistance is a major advantage of blockchain. Once data is recorded in a validated block, it becomes immutable, thus ensuring proof of authenticity. This property is valuable for transaction traceability, combating tampering, and secure archiving.

However, blockchain poses challenges in constrained environments such as MEC: its energy consumption, validation latency (particularly in public blockchains like Ethereum), and

storage management must be optimized. Solutions such as private or consortium blockchains, lightweight consensus algorithms (Proof of Authority, PBFT), or the use of Segregated Witness (SegWit) to reduce block sizes are avenues for adapting to edge computing. By combining blockchain with SDN and NFV, it is possible to create self-managed and secure architectures where flows are dynamically controlled, network functions are deployed on demand, and security policies are embedded in the blockchain, thus promoting transparency, resilience, and trust.

1.2.4 InterPlanetary File System

The InterPlanetary File System (IPFS) is a distributed file system that aims to decentralize data storage. It is based on a peer-to-peer architecture where each file is identified by a cryptographic hash, thus guaranteeing its integrity. Unlike traditional location-based addressing systems (e.g., URLs), IPFS adopts content-based addressing, making data immutable and verifiable.

The functioning of the IPFS is based on a decentralized and distributed approach to file management. Unlike the traditional web, where a file is located by its address (URL) on a central server, IPFS identifies and locates data according to their content. Each file added to the network is cut into small blocks and each block receives a unique identifier called CID (content identifier). This CID is generated using an cryptographic hash algorithm, thus guaranteeing that the content cannot be modified without changing its identifier.

Once the blocks have been created and identified, they are distributed through a P2P network. In this model, each participant in the network (called node) can host, share or request files. When a user wishes to access a file, IPFS does not question a central server, but directly research the nodes that have the blocks associated with the requested hash. This makes it possible to recover data from several sources in parallel, which improves the speed and resilience of the system.

The integrity and authenticity of the data is intrinsically guaranteed by the content addressing mechanism. Indeed, if a block is altered, its identifier (CID) no longer corresponds and the file is rejected. This principle prevents falsification and ensures that users always receive the planned content exactly. In addition, the distributed cache system optimizes exchanges by making it possible to temporarily keep the files requested frequently on different nodes.

Figure 1.8 illustrates the operating principle of IPFS through the main stages of the process. The figure highlights the three fundamental principles of IPFS: chunking, content-based addressing, and peer-to-peer distribution, which together provide a decentralized, robust, and verifiable solution for data storage and sharing. First, a file is chunked into several blocks (Block 1, Block 2, Block 3). Each block is associated with a CID, generated from a cryptographic hash function that guarantees the integrity of the content. These blocks are then distributed across different nodes in the peer-to-peer network (Node A, Node B, Node C). Each node can store one or more parts of the file, thus contributing to the system's redundancy and resilience. When a user wishes to retrieve a file, they submit the corresponding CID to the network. The system

then identifies the nodes holding the required blocks and downloads them in parallel. This approach helps improve retrieval speed and ensures content availability even if some nodes become inaccessible.

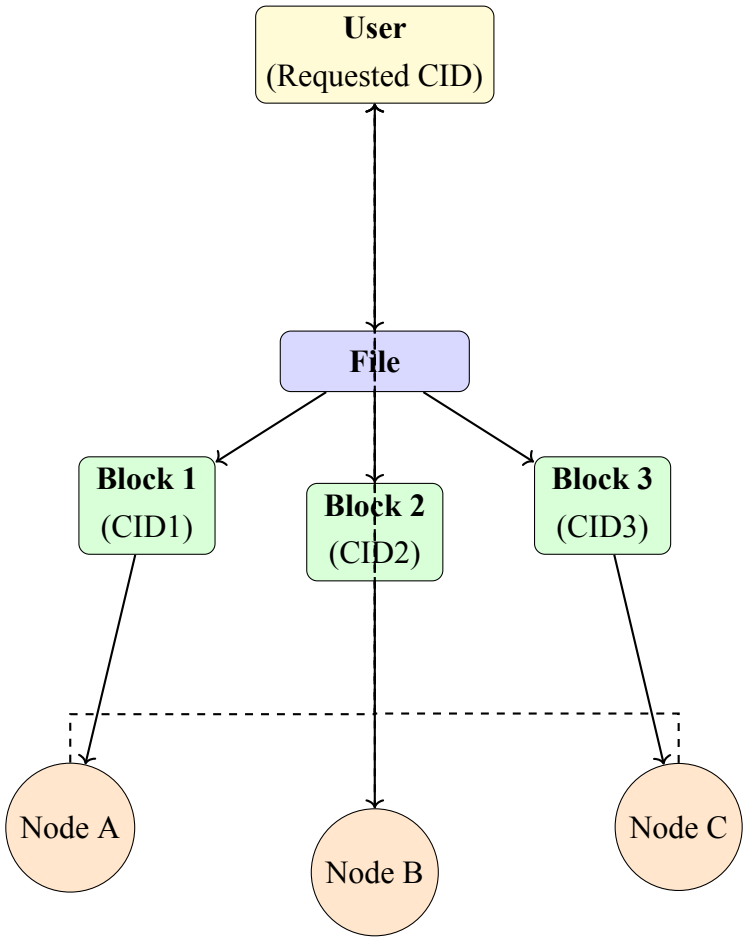


Figure 1.8: Principle of IPFS operation: file splitting, content addressing, and P2P distribution.

In the context of MEC, IPFS provides an effective solution to the need for local, distributed, and secure storage. It allows data to be distributed across multiple edge nodes while ensuring its availability even if a node goes offline. This makes it particularly suitable for video surveillance, IoT data storage, and federated learning scenarios. One of the major advantages of IPFS is the reduction of network load: frequently requested data can be served locally without requiring a request to the cloud. This property improves overall performance and reduces latency, essential criteria for MEC applications.

Combined with blockchain, IPFS enables the creation of a decentralized and verifiable storage system. Files are hosted in IPFS, while their hashes are recorded on the blockchain. When a file is retrieved, its integrity can be verified by comparing the hash with the recorded one. Furthermore, IPFS is scalable and compatible with smart contracts, allowing for the automation of storage and deletion policies, access restrictions, and access tracking. Thus, in a MEC architecture, IPFS plays a crucial role by complementing the computation and communication layers with a distributed, resilient, and secure storage layer, essential for building autonomous

and scalable systems.

1.3 Security in MEC networks

As discussed earlier, the MEC paradigm represents a significant advancement toward low-latency distributed processing, essential for mission-critical applications. However, by bringing computing resources closer to end users and multiplying network entry points, MEC introduces new attack vectors, exacerbating challenges related to data, communications, and resource security. The distributed and heterogeneous nature of MEC requires a fundamental review of traditional security models.

1.3.1 Security policy

One of the major challenges in the field of MEC lies in the lack of uniform and consistent mechanisms for enforcing security policies. Indeed, critical functions such as IP address whitelisting, segmentation via virtual local area networks (VLANs), and secure VPN tunnel termination are often implemented inconsistently or, in some cases, completely neglected [45]. This shortcoming poses a considerable risk to the integrity of the MEC network. On the one hand, it exposes the infrastructure to unauthorized access and lateral propagation of attacks between different network layers and nodes. On the other hand, the lack of consistent control rules makes it difficult to trace traffic flows, limits anomaly detection capabilities, and complicates centralized security policy management. Thus, the lack of effective and standardized implementation of these functions not only compromises the resilience of MEC environments, but also weakens the confidence of users and critical services that rely on this distributed infrastructure.

1.3.2 Edge–Cloud Coordination

Another fundamental challenge of MEC lies in the coordination between the edge and the cloud. The lack of reliable and efficient communication capabilities between these two environments poses a critical problem [46], especially when services require real-time support. In practice, many applications (such as interactive video, connected vehicles, or augmented reality services) require dynamic resource orchestration: some tasks must be processed locally at the edge to minimize latency, while others require greater computing power, available only in the cloud. However, when synchronization, context transfer, or joint management mechanisms for processing instances are not properly implemented, this leads to significant performance degradation. The consequences include increased latency, inconsistent data processing, and a decrease in the QoE perceived by users. More broadly, this lack of coordination weakens the MEC's ability to guarantee continuity of service, particularly for time-sensitive critical applications, and thus limits the large-scale adoption of this hybrid architecture.

1.3.3 Security Related to MEC–Blockchain Synergy

The combination of MEC and blockchain technology is attracting growing interest due to its potential to enhance trust, traceability, and resilience in distributed services. However, the coordination between these two paradigms remains underexplored in large-scale industrial deployments, opening the door to multiple security challenges [47]. On the one hand, blockchain introduces significant computational and energy overhead (e.g., for consensus), which can conflict with the latency and limited resource constraints inherent in MEC environments. However, the integration of blockchain mechanisms into MEC architectures raises complex issues related to data confidentiality, cryptographic key management, and securing communication channels between edge nodes and blockchain networks.

Furthermore, the lack of clear standards and proven models for orchestrating interoperability between MEC and blockchain leaves hidden risks for practical deployments. These risks include: the difficulty of ensuring consistency between blockchain transactions and real-time decisions made at the edge; increased exposure to delayed consensus attacks that can compromise service availability; and threats related to insider attacks when compromised edge nodes simultaneously participate in the blockchain network. Finally, the distributed and decentralized nature of these two technologies, while improving resilience, significantly complicates the implementation of uniform audit, monitoring, and incident response mechanisms. Thus, the MEC–Blockchain synergy offers strong promise in terms of security and trust, but still requires extensive work to identify, classify, and mitigate specific vulnerabilities likely to emerge in real-world deployments.

1.3.4 Security Complexity and Overhead

Another major security challenge in MEC concerns the increasing complexity and overhead associated with deploying multiple protection and learning mechanisms [48]. To address emerging threats, MEC environments often integrate heterogeneous solutions, such as AI-based intrusion detection systems, dynamic access policies, or adapted encryption algorithms to limited resources. However, the layering of these mechanisms results in a significant increase in overhead in terms of computation, energy consumption, and network resource management.

This overhead is not only a performance issue; it can also create new vulnerabilities. The excessive complexity of security systems makes them more difficult to configure and maintain, increasing the risk of human error or integration flaws. Moreover, the simultaneous use of multiple machine learning approaches for attack detection can generate conflicting decisions, slow down the response process, and, in some cases, introduce biases that can be exploited by sophisticated attackers.

Thus, instead of strengthening the resilience of the MEC, the uncontrolled accumulation of security mechanisms can paradoxically weaken the entire infrastructure. To ensure balance, it is essential to adopt a ”security-by-design” approach that prioritizes simplicity, interoperability,

and resource optimization, while ensuring sufficient coverage against threats.

1.3.5 Security of Metadata and Sensitive Data

Protecting metadata and sensitive data is a crucial issue in MEC. Unlike centralized environments where data is stored and processed in highly secure data centers, MEC relies on edge devices often located close to users and exposed to less controlled environments [49]. This proximity increases the risk of compromise: an attacker who successfully infiltrates a MEC node can access not only application data, but also metadata, which reveals critical information about an organization's business, operational workflows, or user browsing behavior.

The loss or exposure of such information can have serious consequences. On the one hand, metadata can be exploited to build detailed user profiles, exposing consumption habits, sensitive locations, or professional relationships. On the other hand, organizational data directly compromises the confidentiality of internal processes and can serve as a basis for targeted attacks, such as industrial espionage or strategic sabotage. The threat is further amplified when compromised devices are integrated into large-scale distributed architectures, allowing attackers to collect and correlate information from multiple network locations.

Faced with this risk, traditional data encryption approaches are insufficient. It is becoming necessary to adopt holistic solutions, including metadata encryption, anonymization of browsing behavior, and strict and contextual access control mechanisms. Furthermore, implementing edge-to-edge intrusion detection strategies is essential to limit the impact of a potential compromise.

1.3.6 Vulnerabilities and Threats Specific to MEC

unlike centralized cloud architectures, edge nodes are often deployed in open environments (e.g., base stations, public Wi-Fi access points), making their physical access easier for malicious actors. These nodes can be exposed to attacks such as:

- Interception of sensitive data,
- Malware injection,
- Distributed denial of service (DDoS) attacks targeting limited resources,
- Spoofing of mobile or IoT devices [120].

Furthermore, user mobility makes it difficult to maintain a persistent secure session, while the variability of the network context (link quality, changing anchor point) complicates the management of certificates, keys, and access control policies [121].

1.3.7 Security Constraints in a Distributed Environment

one of the fundamental challenges of MEC is that traditional security mechanisms based on a centralized infrastructure are not directly transferable. The latency added by a round trip to the cloud for authentication or integrity verification violates MEC performance objectives (latency < 10 ms in some cases) [122]. It is therefore necessary to favor lightweight, distributed, and autonomous approaches, such as:

- Local authentication via certificate or zero-knowledge proof,
- Low-cost symmetric encryption for real-time flows,
- Context-based management of access authorizations (by geolocation, user profile, or application type) [123].

1.4 Limitations of existing solutions

Despite the progress achieved through the integration of advanced technologies in video surveillance systems, existing solutions still face several critical limitations that restrict their effectiveness and scalability. These limitations are mainly observed in three key areas. First, network management and security remain challenging due to the dynamic nature of video surveillance traffic and the increasing vulnerability to cyberattacks. Second, NFV-orchestration, although promising, introduces complexity in coordinating virtualized resources and ensuring reliable service delivery across heterogeneous infrastructures. The adoption of blockchain, while enhancing data integrity and trust, raises new challenges related to scalability, latency, and energy consumption, which are particularly problematic in resource-constrained video surveillance environments.

1.4.1 Network Management and Security

The ETSI GS MEC 003 (Group Specification) [50] proposes a framework and reference architecture for Multi-access Edge Computing. The architecture enables the implementation of MEC applications on based on virtualization infrastructure and close to end users. The hierarchical architecture that vertically divides the MEC into three layers: User layer, Edge layer, and Cloud layer. The user layer is characterized by the wireless communication mode between mobile devices and the wireless infrastructure. The edge layer refers to the computing resources of edge servers. The cloud layer refers to the computing resources of cloud servers. The deployment of MEC servers to process user requests at Small Cell Clouds (SCC) architectures will improve significantly the network Qos and security; optimize intensive data offloading tasks by reducing latency and location-awareness [51]. Mach et al. [52], propose a Small Cell Management (SCM) for controlling SCCs, managing storage and processing resources provided

by SCeNBs. Dynamic and elastic management as SCeNBs can be activated or deactivated by end users at any time. Jararweh et al. [51], propose a MEC architecture based SDN to reduce network management and administration costs. The architecture is composed of three main layers. SDN control layer which contains an edge cloudlet module and a task monitoring to verify, task and process management, resource management, user access control and provide an overview of the network topology. The edge computing layer which contains an Edge Cloudlet node composed of virtual Famtocell BS and Edge Cloudlet to request mobile data and services from SDN module, broadcast mobile data to UE including number of users and energy level of nodes (UE, server). For QoS guarantees, Yang et al. [53] propose an Open RAN software defined architecture through a virtualization infrastructure to administrate, configure and provide cool programmable interfaces to mobile network. The proposed architecture include three main layers : Wireless Spectrum Resource Pool (WSRP), Cloud Computing Resource Pool (CCRP) and SDN controller module. Phan et al. [54], propose an SDN-based fog service of dynamic data offloading among fog nodes for a intensive computation tasks. This service made it possible to select an optimal offloading node and control the offloading path using SDN controller, while guaranteeing a high Qos in terms of bandwidth and end-to-end delay. Elgendy et al. [55], control the intensive tasks of data offloading by energy parameters. They propose a formal model integrating the offloading computation, security and resource allocation as an optimization problem. The goal is to reduce the overall time and energy when data offloaded from mobile devices. The simulation results demonstrates that the proposed model reduces significantly the offloading overhead by 64.7 % compared with local execution approach. Hao et al. [56], propose an energy-aware heuristic task scheduling technique for transferring wirelessly energy among edge nodes and manage VM migration. The proposed solution is based on several green mobile sources such as solar energy to extend the life time of edge nodes in case of intensive offloading. After clustering, the head of cluster can initiate and manage VM migration, perform offloading plan and allocate tasks.

For MEC application security, a multi-layered distributed SDN-IoT framework based on Blockchain is proposed in [57], that provides efficient cluster head selection and secure network communication through route switch identification and isolation. The proposed Blockchain Security for SDN (BSS) in [58] protects privacy and availability of resources against non-trusting nodes. The authors simulate the proposal using OpenStack framework, where the results demonstrate resilient Blockchain security over distributed peer to peer systems.

Although SDN and Blockchain technologies have individually attracted great intention from industry and academia in recent years, secure mobile network based SDN in MEC environments still remains a challenge, where not further defined so far. This warrants us in this chapter to propose MEC architecture based SDN for secure network communication and secure Edge infrastructure. Table 1.2 below resume our review of MEC application features and required QoS across several IoT domains. However, from the table, there is a lack of research work that presents a model format that can meet the requirements and future of MEC applications.

Table 1.2: MEC applications features and Required QoS.

Application domain	MEC Apps	Features of MEC	Required QoS	Approach
Intelligent Transportation Systems (ITS)	• Smart Parking • Vehicle to Everything Communication (V2X) • Internet of Vehicles (IOV) • Social Internet of Vehicles (SIoV) • Remote control • Self-driving	• Mobility Support • Dense geographic distribution • Heterogeneity • Ultra-low latency	• Dissemination of information in real time. • Reliability • Ultra-high bandwidth speed (5G)	• Clustering [59, 60] • Orchestration [61] • Blockchain in vehicular messaging [62, 63] • Privacy frameworks and Protocols [64] • Adaptive genetic approach [65]
Resources management	• Allocation of resources • Resource aggregation • Control energy consumption • Management of UE requests • Managing heterogeneity • Decision to unload mobile codes	• Sensitive to Latency • Heterogeneity • Proximity • Dense geographic distribution	• Low latency • Optimized use of resources	• Lightweight virtualization [66, 67] • Lightweight codes unloaded [68] • Active resource control [69] • Nonlinear optimization [70]
Security and Privacy	• Intrusion detection • Authentication • Confidential access • Prevent attacks	• Sensitive to Contexts • Dense geographic distribution • Location-aware	• Availability • Authenticity • Privacy • Data integrity	• Blockchain (tamper-proof, redundant, self-healing) [71] • Quantum cryptography [72]
Internet of Things (IoT)	• Management of raw data streams • Edge IoT platforms	• Mobility Support • Dense geographic distribution • Heterogeneity	• Minimum End-to-End delay • Energy-insensitive devices	• Proxy migration (VM) [73] • Hierarchical network architecture [74]

1.4.2 NFV-Orchestration

The idea of using an MEC architecture for security management in distributed networks has been addressed in several research works. Similar architectures have been proposed for resource management, NFV, and load balancing in MEC environments. For instance, Hui et al.[75]

introduce a new resource allocation mechanism in MEC environment to ensure the safe and stable operation of the system. The proposed mechanism is based on deterministic differential equation model to tackle the challenge of the resource-constrained network devices. Fengxian et al., [76] propose a Blockchain based Mobile Edge Computing (B-MEC) framework for adaptive resource allocation and computation offloading in future wireless networks. The main goal of this framework is to tackle the challenge of the consensus between the edge nodes while simultaneously guaranteeing the performance of both MEC and blockchain systems. In which, resource allocation, block size, and the number of consecutive blocks are critical to the performance of the proposed B-MEC. Elgendy et al., [77] presents a multi-user resource allocation and computation offloading model with data security to address the limitations of such devices. The proposed model is based on optimal computation offloading algorithm to determine the optimal offloading decision for the mobile users (MUs). Managing security in these environments also represents a significant challenge. Many research studies have looked at different approaches to ensuring data and communications security in settings where resources are limited and connectivity may be intermittent. The advantages of the Edge computing can enhance security and greater improve scalability due to the fact that any attack on an edge server will affect the users connected to that server and not the whole network [78]. To meet the standard requirements of MEC, Debaak et al.,[79] propose a seamless secure anonymous authentication scheme (S-SAAS) to create a secure session in cloud-based MEC. Similarly, Huang et al., [80] propose a security overhead model which takes into account the influence of different performance parameters such as the size of protected dataset, the CPU cores, computation frequency of MEC servers.

The orchestration and resource allocation mechanisms also play a crucial role in improving the overall performance, security and economic efficiency of an MEC network by enabling dynamic and efficient management of available resources. Xiantao et al., [81] discuses the resource allocation issues for MEC-based video streaming. They review the network architecture, the resources management, the optimization criteria and tools. Zhang et al., [82] propose an efficient technique to balance the loads among base stations with security layer while reducing the system cost. They incorporate an Advanced Encryption Standard (AES) cryptographic strategy suffused with encryption and decryption keys contrived from electrocardiogram (ECG) signals to secure devices during different stages of data transmission. Liu et al., [83] propose a novel blockchain-based framework with adaptive block size for video streaming with MEC. To avoid the security overload caused by bolckchain nodes, they propose a block size adaptation scheme with two offloading modes: offloading to the nearby MEC nodes or a group of device-to-device (D2D) users. Hiroki et al., [84] present a MANO (Management and Network Orchestration) for deploying services composed of chained Basic fonctions (BFs) with requirements to computing and network resources of distributed nodes. They propose a heuristic method for calculating the optimal placement of BFs to make the edge network more efficient for system overload like data security. Quan et al., [85] propose a multi-agent-based

cross-domain resource orchestration framework to meet the challenges of jointly optimizing information flow and traffic flow.

The table 1.3 provides an overview of the security features of MEC applications along with their associated security policies. These security features encompass robust authentication policies, data encryption practices, access control mechanisms, measures to ensure data integrity, environment isolation techniques, threat monitoring strategies, and network security protocols.

Table 1.3: MEC application features and required security policies.

Security feature	Description	Security policy
Robust authentication	Implementation of strong authentication mechanisms to verify the identities of users and devices.	• Management of keys and signatures [86]. • Employment of authentication protocols [87, 88].
Data encryption	Use of encryption techniques to ensure confidentiality.	• Application of Homomorphic Encryption [89]. • Implementation of auditing and monitoring practices [90].
Access control	Management of access authorizations to resources and data based on predefined roles and policies.	• Implementation of Authorization mechanisms [91]. • Utilization of Certification-based access control [92].
Data integrity	Implementation of mechanisms to detect and prevent unauthorized alteration or tampering of data.	• Verification scheme [93]. • Use of Aggregate signature techniques [94].
Isolation of environments	Logical and physical separation of resources to prevent breaches of data confidentiality and integrity.	• Implementation of Virtual private network (VPN) solutions [95].
Threat Monitoring	Continuous monitoring of activities to detect suspicious behavior or intrusion attempts.	• Deployment of Distributed intelligence mechanisms [96].
Network Security	Implementation of measures to protect against network attacks such as denial of service (DDoS) attacks and data interception.	• Utilization of Blockchain-based security protocols [97]. • Secure mobile data offloading[98]. • Blockchain based privacy preserving [99].

1.4.3 Blockchain and security challenges in video surveillance

The integration of blockchain technologies and next-generation architectures in VSS has attracted considerable interest in academic research. This convergence of technologies promises to address various aspects of security, privacy, data integrity, and resource management in video surveillance environments. To achieve these objectives, it is imperative to address several security and privacy challenges inherent in these complex areas, namely privacy content protection [100], data integrity verification [101], detection of falsifications [102], resource allocation [103], as well as transmission and delivery models [104]. It is also essential to note that the generated data by VSS can be considered in resource-constrained networks [105]. Depending on the security and communication requirements of these applications, the use of traditional blockchain solutions in these contexts may decrease the effectiveness of the security approach due to increased resource requirements (such as storage, computation, bandwidth, etc.) needed to scale with the blockchain size. Furthermore, the use of Proof of Work (PoW) can negatively impact the overall network performance by increasing latency, as noted in the work [106].

To avoid this scalability problem in resource-constrained networks while maintaining network performance, several studies have considered the integration of MEC architectures to reduce security overhead. Wang et al. [107], address the issue of blockchain scalability when processing video segments captured by a video surveillance system. They have developed a blockchain solution specifically suited to an edge computing environment, which leverages IPFS technology to provide large-scale storage capacity within a dedicated infrastructure. Similarly, Najmath et al. [108] have developed a decentralized storage solution for video surveillance designed in a mobile edge environment. The proposed architecture is developed to generate video evidence based on automatic source identification and metadata stored in the blockchain. Additionally, many proposals aim to reduce the blockchain scalability overhead by substituting Proof-of-Work (PoW) mechanism with authentication process, clustering techniques, and introducing a multi-level hierarchy [109]. Therefore, it is not recommended to remove blockchain fundamentals, as this could compromise the overall effectiveness of the security offered and potentially lead to security vulnerabilities.

EAMSuS [110], Efficient Algorithm for Media-based Surveillance System, is one such approach that enhances media-based surveillance within IoT smart city frameworks. This algorithm focuses on improving the accuracy of event detection and the efficiency of video analysis by optimizing processing times. EAMSuS is particularly effective in real-time surveillance applications, making it a valuable component in smart city infrastructures. However, while EAMSuS excels in detection accuracy and processing speed, it does not specifically address the optimization of storage and transmission efficiency, which are critical for large-scale deployments. The video forensics framework proposed in [111], provides a robust solution for maintaining the integrity and authenticity of video evidence using blockchain technology. This

framework ensures that video data remains tamper-proof, making it particularly suitable for legal and security applications. By providing reliable verification of video authenticity, the Video Forensics Framework addresses the critical need for trustworthy video evidence. However, its primary focus is on forensic capabilities rather than optimizing network performance or scalability for general video surveillance. BlockSee [112] is a blockchain-based system designed for IoT video surveillance in smart cities, providing a secure and immutable record of video data. By utilizing a permissioned blockchain and smart contracts, BlockSee ensures that video transactions are validated and stored immutably, preventing unauthorized tampering or alterations. The system is highly distributed, enhancing data security and integrity across a smart city infrastructure. However, while BlockSee effectively secures video data, it may face challenges in optimizing network performance and scalability for large-scale deployments. The proposed video integrity verification method in [113], leverages blockchain technology to ensure the authenticity and integrity of video data. By creating an immutable and verifiable record of video transactions, this approach prevents unauthorized alterations, making it suitable for applications where video authenticity is crucial. However, this system may not fully address network performance optimization and scalability challenges associated with large-scale video data. Kumar et al. [114], provide a comprehensive survey of passive video forgery detection techniques, reviewing various methods to detect tampering and alterations in video data. While the focus is on enhancing the detection of forged videos to ensure content reliability, this survey does not address the integration of these techniques with blockchain or edge computing for improved network performance and scalability.

The personalized blockchain proposed by Sohaib et al., [115] is specifically designed to adapt to networks with limited resources while addressing concerns related to energy efficiency and security. Rather than adopting a proof-of-work (PoW) algorithm, the authors suggest the use of an efficient access control model for this network. This model helps reduce network congestion by implementing a distributed authentication mechanism managed by an SDN controller within a cluster structure. The objective of the public blockchain outlined in the work [116] is to address energy conservation concerns in such context. This is achieved by simplifying the security aspects of the blockchain and mitigating network congestion. Notably, the authors have replaced the Proof of Work (PoW) mechanism with an authentication approach involving a cluster of private blockchain nodes through the SDN controller. Despite the adoption of security solutions mentioned previously for resource-constrained environments, it is not recommended to completely remove any key concepts, as this could compromise security effectiveness. Additionally, these solutions are likely to have security vulnerabilities due to the level of centralization of control, which remains a critical factor to consider.

Unlike previous research, Yiming Liu et al., [117] developed a Blockchain-based MEC architecture to solve the problem of video processing in a decentralized way, adapting to increasing user demands. This approach differs from previous architectures because it preserves the efficiency of blockchain while eliminating the centralization present in SDN architecture.

This is accomplished through the execution of a set of smart contracts and the replacement of Proof of Work (PoW) with Proof of Stake (PoS). Another blockchain-based video streaming transcoding MEC architecture was proposed in [118]. The authors consider the blockchain scaling process as a distributed ADMM optimization problem. In particular, a smart contract is implemented between all parties involved in order to solve the distributed optimization problem. In the next chapter [119], we will propose a secure MEC architecture based on SDN and BC to ensure secure mobile data offloading and reduce latency caused by blockchain overheads on Small Cell Networks (SCN). Each mobile user can transfer their data directly to a central cloud with a public blockchain if connected via cellular networks; otherwise, they can transfer indirectly to a private BC via complementary networks (Wi-Fi or D2D). The effectiveness of blockchain is evident by enhanced DPOS (delegated proof of stake) consensus to ensure the authenticity of transactions transferred to a public blockchain. Although these models are consistent and ensure a high level of data security and efficiency, they are not simplified and do not guarantee a trade-off with network performance.

The integration of SDN and NFV enhances security by enabling the implementation of dynamic policies through a centralized controller with a holistic view of the network. This architecture supports mechanisms such as micro-segmentation, the isolation of suspicious traffic, and the redirection of packets to virtualized intrusion detection and prevention systems (IDS/IPS) [124]. In parallel, NFV facilitates the on-demand deployment of virtualized security functions (e.g., vFirewall, vIDS, vVPN), which proves particularly advantageous during overload conditions or incident response. Moreover, NFV orchestrators (NFVO) are capable of identifying anomalies in the behavior of these virtualized functions themselves [125].

Blockchain technology offers a promising framework for ensuring distributed identity management, data integrity, and transaction traceability across heterogeneous environments. The integration of smart contracts further strengthens this paradigm by enabling the automated enforcement of predefined security policies, thus ensuring consistency and reliability across decentralized infrastructures. In particular, the use of smart contracts allows for the automatic enforcement of security policies such as:

- Revocation of a compromised terminal,
- Auditing access to sensitive data,
- Secure negotiation of sessions between edge nodes and users [126].
- Variants such as Hyperledger Fabric (consortium blockchain) or IOTA (directed acyclic graph) are better suited to resource-constrained environments such as the MEC [127].

IPFS for distributed storage security combined with blockchain, IPFS can ensure that locally stored data has not been tampered with. When retrieving a file, its hash is compared to the one stored in the blockchain. This ensures content integrity, which is essential for critical data such

as system logs, surveillance videos, or configuration files [128]. For emerging approaches, promising avenues include:

- The application of machine learning for edge intrusion detection, capable of learning normal patterns and detecting deviations in real time [129].
- Zero-Trust architectures, where every request, even from a ”legitimate” endpoint, must be dynamically authenticated and authorized, without assuming implicit trust [130].
- Homomorphic security allows computations to be performed on encrypted data even in not fully trusted MEC nodes [131].

Conclusion

In this first chapter, we explored the theoretical and technological foundations of MEC, highlighting its transformative role in next-generation networks. MEC represents an ambitious response to the limitations of centralized cloud architectures by bringing processing, storage, and control resources closer to the network edge. This approach promotes significant latency reduction, optimized bandwidth usage, and improved QoE for time-sensitive applications such as telemedicine, 4K video, connected vehicles, and smart city services.

We highlighted the enabling technologies of MEC, including SDN, NFV, Blockchain, and IPFS, which together form a dynamic, programmable, resilient, and secure infrastructure. SDN enables intelligent and centralized management of network flows; NFV ensures flexible functional deployment; blockchain introduces a distributed and tamper-proof layer of trust. While IPFS provides an efficient and verifiable decentralized storage model, their seamless integration is essential to overcome the challenges inherent in MEC environments.

However, these technical promises come with complex issues, particularly regarding QoS and security. MEC, due to its distributed, mobile, and heterogeneous nature, introduces new vulnerabilities: physical exposure of edge nodes, resource limitations, uncertainties regarding communication reliability, and the absence of a fixed security perimeter. Protecting data, users, and services therefore requires distributed, adaptive, intelligent, and lightweight approaches.

Through analyzing threats specific to MEC, we identified the need for solutions focused on automating security policies, proactive intrusion detection, trust without a central authority, and resilience to local failures. Security cannot be conceived as a single layer, but must be considered as a multi-level strategy, ranging from the protection of IoT devices to service governance and network orchestration.

This chapter therefore laid the conceptual foundations, and limitations of current solutions, necessary for understanding the challenges of our research. Chapter 2 will therefore focus on proposing a secure data offloading architecture in small cell networks, leveraging the synergy between SDN and Blockchain to meet security and QoS requirements in distributed environments with high mobility.

Chapter 2

Securing data offloading in small cell networks

Chapter 2

Introduction

Over the last decade, mobile computing (MC) has experienced a profound evolution with the emergence of MEC. This paradigm shift is motivated by the limitations of traditional cloud architectures, which cannot efficiently support the stringent requirements of modern IoT applications. Conventional cloud models often suffer from high traffic latency, limited real-time responsiveness, and security vulnerabilities, making them unsuitable for applications that demand Ultra-Reliable and Low-Latency Communication (URLLC), such as autonomous driving, smart healthcare, and large-scale video surveillance. To address these shortcomings, SDN and blockchain technologies have gained significant attention. SDN introduces programmability and centralized control to optimize network resource management and adapt to dynamic traffic conditions, while blockchain provides decentralized trust mechanisms that reinforce data integrity, transparency, and security. Despite these advancements, resource-constrained mobile devices still face difficulties in meeting security and performance requirements without relying on data offloading mechanisms.

In this chapter, we present a secure architecture for mobile data offloading in Small Cell Networks (SCNs), where MEC plays a pivotal role in reducing latency and enhancing computational efficiency. The proposed architecture integrates SDN for intelligent traffic management and blockchain for decentralized security enforcement. More specifically, each mobile device is able to offload its data either directly to a central cloud through a public blockchain, or indirectly through Wi-Fi and D2D communication within a private blockchain environment. This dual-path strategy leverages both public and private blockchain infrastructures to balance security guarantees, network efficiency, and resource availability. The evaluation of the proposed framework demonstrates its ability to maintain ultra-low latency, while simultaneously ensuring robust data security and efficient resource utilization in MEC-based SCNs. This work highlights the potential of combining SDN and blockchain technologies to enable trustworthy, scalable, and high-performance mobile data offloading solutions.

2.1 Secure Multipath data offloading architecture in MEC

In this section, we introduce the proposed secure multipath data offloading architecture designed for MEC environments. The architecture, illustrated in Figure 2.1, is structured as a multilayer framework for SCNs. It is composed of four distinct layers: the Core Layer, the Control Layer, the SDN Layer, and the Radio Access Network (RAN) Layer. The latter integrates diverse edge networks, including small cells and D2D networks, to ensure flexible and efficient data offloading. This layered design enables the separation of concerns, where each layer is assigned specific roles in managing computing, storage, and communication resources. The architecture aims not only to enhance security and trust through blockchain integration, but also to guarantee low-latency performance, efficient resource orchestration, and adaptive scalability in dynamic MEC scenarios.

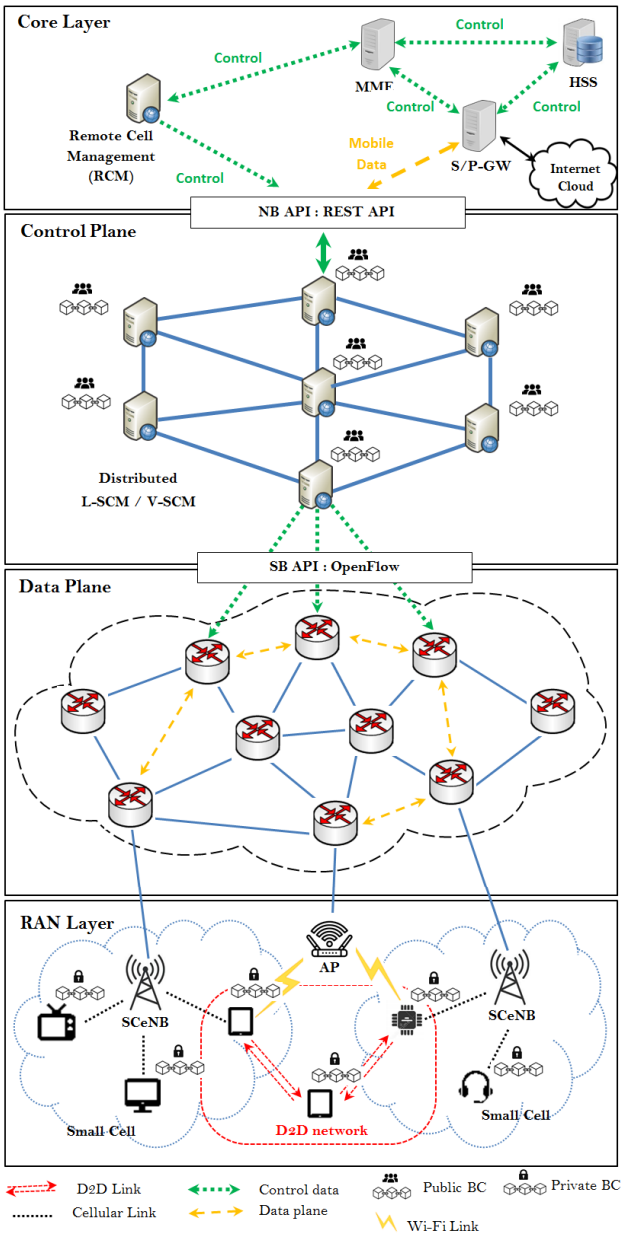


Figure 2.1: Secure Multipath data offloading of the Small Cell Networks (SCN).

2.1.1 Core Layer

The Core Layer represents the highest system level of the proposed architecture. Its primary objective is to extend and enhance the capabilities of SCCs by integrating advanced computing and storage resources. Within this layer, several functionalities are consolidated to ensure the reliability and efficiency of MEC-based SCNs:

- *Management of SCeNB nodes:* The Core Layer provides seamless management of Small Cell enhanced NodeBs (SCeNBs), ensuring their smooth integration into the MEC ecosystem.
- *Elastic resource allocation and aggregation:* Through the adoption of Network Function

Virtualization (NFV) technologies, this layer enables dynamic allocation and aggregation of resources, in line with MEC standards. This improves flexibility and adaptability to fluctuating traffic demands.

- *Remote Cell Management (RCM):* SCeNBs can be activated or deactivated in real-time depending on traffic conditions, reducing energy consumption and improving the sustainability of the network.
- *Support for LTE architectural entities:* The Core Layer incorporates essential LTE components such as the Home Subscriber Server (HSS), Mobility Management Entity (MME), and the Serving/Packet Gateway (S/P-GW). This integration ensures compatibility with existing LTE infrastructures while extending cloud computing services closer to the edge.

By combining these capabilities, the Core Layer significantly enhances dynamic MEC operations, bringing cloud-level services closer to end-users, and enabling a highly responsive and secure data offloading environment.

2.1.2 Control Plane

To address the security challenges of application offloading and the limitations of centralized SDN architectures, the proposed framework integrates a distributed SDN control system reinforced by a public blockchain network. This design ensures that security, scalability, and flexibility are preserved in dynamic and heterogeneous MEC environments. The main objectives of the Control Layer are threefold:

- *Secure multipath offloading flow management:* This layer guarantees secure control of data flows between IoT devices connected through the RAN as well as complementary access technologies, such as Wi-Fi and D2D communication.
- *Optimization of distributed SDN control:* By distributing the control functions across multiple SDN controllers, the architecture mitigates network overload within the MEC layers, thus reducing latency and improving system responsiveness.
- *Secure cloud data storage:* It provides trusted and tamper-proof mechanisms for managing and securing data stored in SCCs, leveraging blockchain to guarantee integrity and authenticity.

2.1.3 Data Plane

This layer deploys distributed OpenFlow switches (OFS) to provide offload paths (i.e. forwarding) based on a control objective. Thus, this layer interacts with the upper layer (Control plane) via

open OpenFlow interfaces. This agile coupling allows the functions of the control plane to be aggregated, as well as providing a global view of the topology and behavior of the network. The OpenFlow protocol makes programmable data planes available to network administrators and network operators via software entities. The agile coupling between the SDN layers translates simplicity of implementation of the control planes as well as the passage of the new rules to the switches. This adds many benefits to our architecture by releasing mobile device dependency to platform owners, effectively securing transmissions, reducing installation costs, removing network overhead, reducing latency that depends generally to the use of the hardware as well as the behavior of the protocols.

2.1.4 RAN Layer

The radio access layer represents the wireless interface of the mobile environment and plays a crucial role in ensuring direct connectivity between end-user devices and the core network (CN). It constitutes the entry point through which devices such as smartphones, tablets, smart TVs, and other IoT-enabled mobile devices establish communication with the MEC-SDN architecture. In the context of the proposed architecture, the RAN is designed to cover a large number of small cells, thereby extending network coverage and improving service availability for dense urban environments and highly mobile users. These small cells serve as the primary access points for user equipment (UE), providing low-latency connectivity and localized data offloading capabilities. The RAN layer integrates multiple communication modes to ensure flexibility and scalability:

- Traditional backhaul links connect small cells to the core network, guaranteeing stable connectivity and facilitating centralized service access.
- Wireless links (Wi-Fi) offer an alternative and cost-effective means of offloading data traffic, alleviating congestion in cellular bands.
- Direct D2D communication enables nearby devices to exchange data without relying on intermediate infrastructure, thereby reducing latency, enhancing spectral efficiency, and improving network resilience.

By leveraging these complementary communication paths, the RAN Layer enhances the adaptability and robustness of the proposed system. It ensures that mobile users can benefit from seamless connectivity, regardless of their location or the density of the network. Furthermore, the RAN layer plays a pivotal role in enabling secure multipath offloading, as it provides the physical and logical foundations for forwarding data across diverse access technologies while maintaining the security guarantees established by the upper layers. The RAN Layer acts as the frontline of connectivity in the proposed secure multipath offloading architecture. Its integration of small cells, Wi-Fi, and D2D communications ensures efficient, scalable, and low-latency

access, which is fundamental to supporting the stringent requirements of modern MEC-enabled mobile environments.

2.2 Trust data offloading management and control using Blockchain and SDN

This section explains the mechanisms adopted to ensure secure management of offloaded data across the Radio Access Network (RAN) layer in a distributed manner. Specifically, data management is supported through a combination of distributed SDN controllers organized as a P2P network, a public blockchain operating at the control plane layer, and a private blockchain deployed at the RAN layer. In this framework, SDN controllers trigger the activation of the public blockchain whenever the RAN establishes a connection between UE and the CN. This approach effectively mitigates the security vulnerabilities typically associated with centralized management in legacy network architectures. By leveraging P2P structures reinforced with blockchain, the proposed solution achieves higher levels of security and trust, as demonstrated in multiple experimental scenarios. This observation strongly motivates the integration of advanced blockchain concepts into the distribution of control functions among SDN controllers.

Unlike conventional models that rely on relay nodes to secure communication, our architecture eliminates this requirement. The global network view, inherently shared among distributed SDN controllers, ensures secure interactions without depending on potentially vulnerable intermediate nodes. This design significantly reduces the risk of compromised communication paths. To consolidate these benefits, our system employs both public and private blockchains, each operating under the principles of Distributed Ledger Technology (DLT). In this context, data transmissions between nodes are treated as transactions, which are securely stored, grouped into blocks, and validated sequentially. Each newly validated block is appended to the chain of previously confirmed blocks, preserving immutability and ensuring end-to-end trust in offloaded communications.

2.2.1 Offloading with public Blockchain

A public blockchain network is activated when SDN controllers redirect the network traffic flow from mobile devices to the core network (as shown in the figure 2.2). In this case, each SDN controller of distributed P2P network is connected to small cell network with approved public blockchain by mobile operators. The public chain of the newly connected small cell network generates the first block using stored and committed transactions between distributed SDN controllers. Indeed, each SDN controller can have their own public and private keys to associate to the public BC, authenticate the offloading devices, control continues connectivity and mobility of the devices among cells, isolate malicious offloading paths, storing approved offloading rules and policies. Moreover, the SDN controllers may be involved in consensus

processes to validate offload paths, but with an increased number of transactions, these processes require sufficient computational resources to validate all transactions and maintain a large-scale distributed ledger. In particular, due to the limited resources of IoT devices in MEC architectures, the Proof-Of-Work (POW) consensus processes are not suitable for a large-scale distributed ledger.

To overcome this issue, we propose an improved Delegated Poof-Of-Stake (DPOS) consensus to ensure offloading transaction authenticity, reduce complexity and mobile security over- head in public blockchain. The SDN controllers participating in block validation must be voted by the nodes of the small cells and should incentivized to engage in offloading tasks security, which usually takes the form of rewards. In addition, it prevents any kind of centralized control of data offloading, in other words, guarantees the total decentralization of data offloading. Each controller correctly validates an offloading path or part of this path has a contribution score. This score is determined by the number of offload transactions, the number of blocks and the duration of the offloading, time to have rewards and the transfer rate. The delegators compete to ensure the following rule: "*the more a mobile node sends offload streams to the core network using public BC, the more important the cell controller becomes*". Otherwise, the blockchain network weeds out and isolates controllers that highlight vulnerable aspects of the system for attacks.

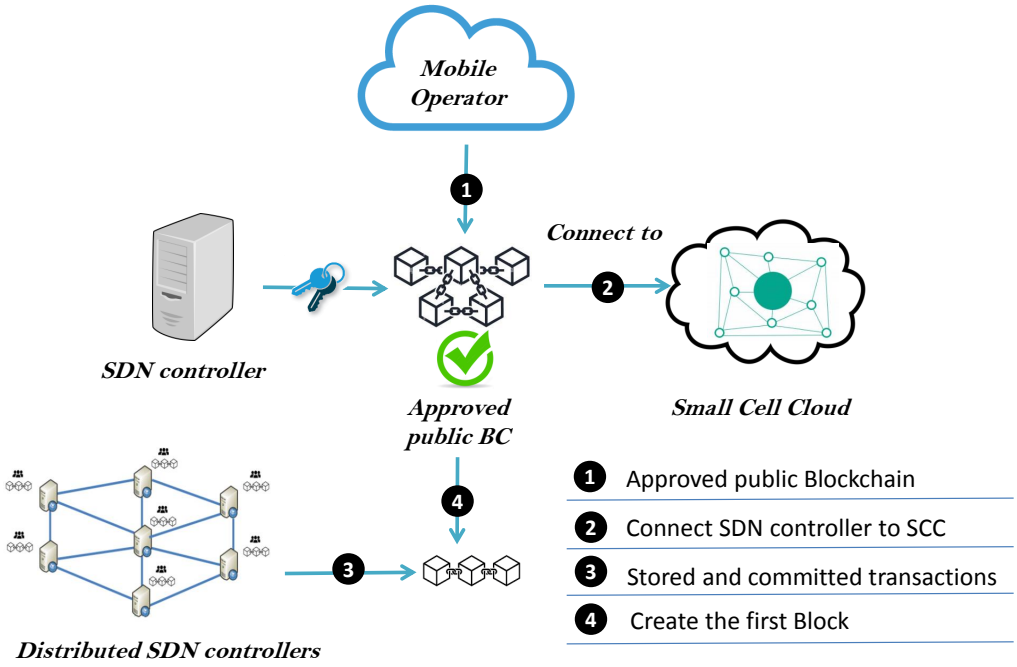


Figure 2.2: Public BC for data offloading

2.2.2 Offloading with private Blockchain

In our architecture, the private blockchain (as shown in the figure2.3) is enabled in each SDN domain to manage and control the mobile devices of each Small Cell by SDN controllers. Each

user device must have permission to participate in network security. The SDN controller is autonomous in choosing which node should participate in a private blockchain network, as participation is restricted to public access. In addition, it uses a set of authentication rules and methods to verify each device identity, efficiently validate its connections, and restrict access to data that are not relevant to them. In our architecture, the private blockchain records the traceability of all communications between nodes, including local communication between IoT devices or remote communication between IoT devices and remote cloud services. To improve latency and increase the efficiency of the mobile network, each IoT device is free to switch its connectivity with networks that suffer from heavy overloads by offloading its data through other communication interfaces available to them. This allows creating alternative paths enabling offloading through D2D networks and avoids tendencies of public/private BC to have longer validation times for new offloading.

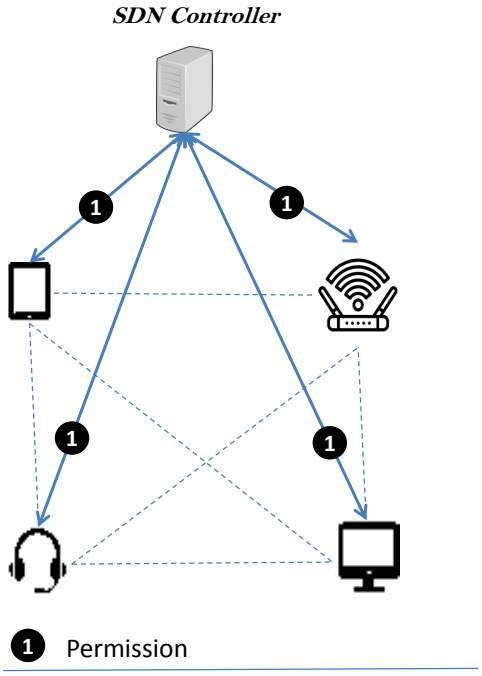


Figure 2.3: Private BC for data offloading

2.3 Evaluation and Simulation Results

The performance evaluation of small cell data offloading with the integration of a private Blockchain has been carried out using a set of tools and platforms that allow realistic simulation and testing of the proposed architecture. The aim of this evaluation is to assess the feasibility, scalability, and efficiency of the proposed model under different networking and IoT scenarios.

The first component of our experimental setup is the Mininet-WiFi emulator [132]. Mininet-WiFi is an open-source network emulator that enables the creation of instant and realistic

virtual network topologies. It provides an efficient platform for developing and testing SDN scenarios with support for the OpenFlow protocol. Additionally, it allows the emulation of IoT applications, wireless environments, and lightweight virtualization software architectures. All of these can be deployed on a single physical machine or distributed across a group of virtual machines. This flexibility makes Mininet-WiFi particularly suitable for simulating our proposed small cell offloading scenario, as it provides the possibility to model wireless nodes, mobility, and diverse connectivity patterns within the RAN. For the Blockchain component, we employed the Hyperledger Fabric platform [133], an open-source project hosted by the Linux Foundation. Hyperledger Fabric is designed to build permissioned ledgers for Blockchain applications. Unlike public blockchains, Fabric provides a higher level of data security, privacy, and scalability, making it especially well-suited for enterprise-level use cases. In our architecture, Hyperledger Fabric serves as the backbone of the private Blockchain deployed within the RAN layer. This integration ensures secure transaction management, data integrity, and fine-grained access control, all of which are crucial for reliable offloading of sensitive IoT data.

The evaluation was conducted on an Ubuntu Virtual Machine (VM) configured with an Intel Core i7 dual-core processor and 8 GB of RAM. This setup was sufficient to support the deployment of both the RYU SDN controller and the Blockchain services. The RYU controller was executed directly on the VM through Python scripts, while the Blockchain peers were instantiated using Docker containers. This hardware-software integration allowed us to emulate the interaction between the SDN control plane and the Blockchain infrastructure in a controlled yet realistic environment.

The simulations considered a representative IoT-enabled SDN scenario, as illustrated in Figure 2.4. The network topology was designed using OpenFlow Switches (OFS) implemented within Mininet-WiFi through a custom Python script. These switches form the foundation of the virtual IoT network and are remotely connected to the RYU controller running on the Ubuntu VM. This configuration makes it possible to simulate the flow of data across the RAN, the offloading process toward the core network, and the role of the Blockchain in securing and validating the transmitted transactions. Through this setup, we are able to test the performance of the proposed architecture under various load conditions, communication patterns, and offloading scenarios, while ensuring reproducibility and scalability of the results.

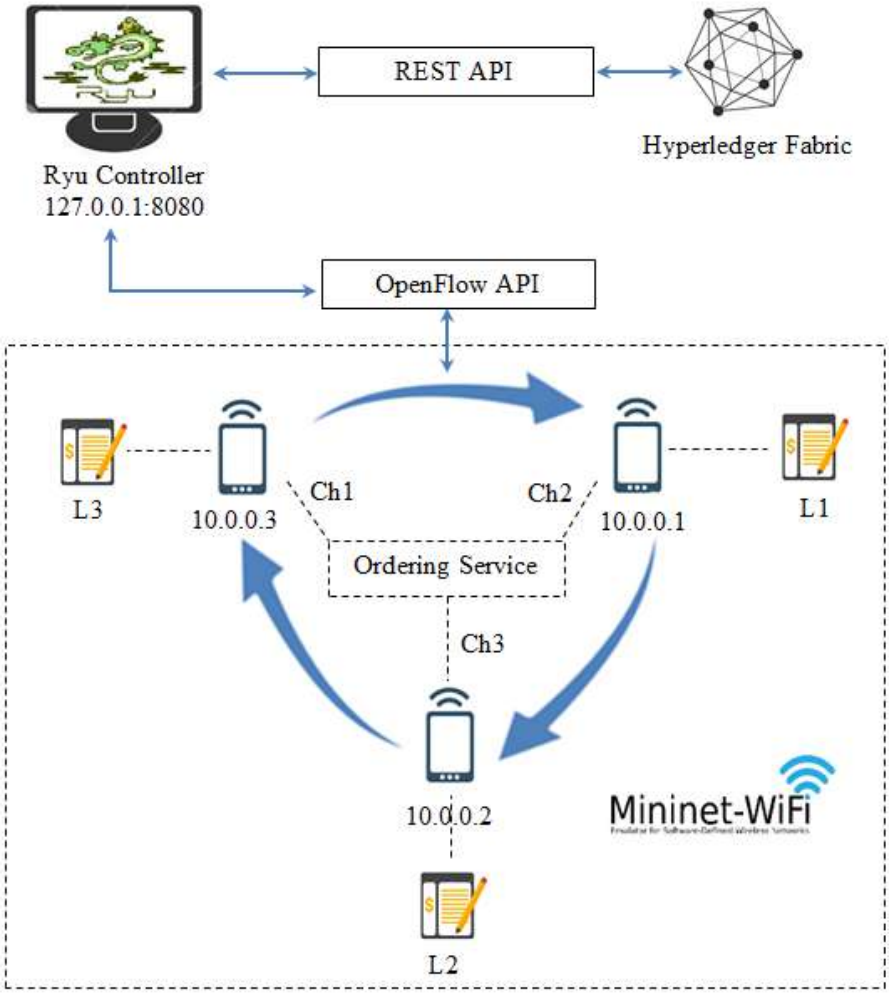


Figure 2.4: MininetWifi network with private Hyperledger Fabric Blockchain.

In our proposed architecture, the devices located within each small cell are integrated into the Hyperledger Fabric Blockchain network and represented as distinct organizations. To illustrate this integration, we consider a small cell (denoted as SC1) that consists of three mobile devices (labeled h1, h2, and h3), an access point (AP1), and an OpenFlow switch. The OpenFlow switch provides connectivity to the Software-Defined Networking (SDN) domain and is managed by the Ryu controller. This setup enables us to model both the networking and Blockchain layers within a unified framework.

The Fabric Blockchain network deployed for this scenario includes three peers (P1, P2, P3),

three mobile devices (h1, h2, h3), an ordering service, as well as blockchain ledgers (L1, L2, L3), chaincode (smart contracts), and channels (Ch1, Ch2, Ch3). Within this configuration, the mobile devices interact with the Blockchain peers as client nodes. More specifically, user applications running on the devices are able to initiate transactions and issue them to the corresponding peers (P1, P2, and P3). Each Blockchain peer is configured with endorsement policies that define the rules for validating transactions. The validation process relies on chaincode, which in this case is implemented using the Go programming language. When a transaction is initiated by a client node, it is first endorsed by the relevant peers based on the chaincode logic. Once endorsed, the transaction is cryptographically signed and forwarded to the ordering service. The ordering service plays a critical role in the Blockchain workflow: it authorizes the creation of new blocks, manages the organization of transactions into blocks, and establishes communication channels (Ch1, Ch2, and Ch3) between the small cell and the device-to-device (D2D) organizations.

After the ordering phase, the validated transactions are dispatched to the committing peers, which are responsible for updating the distributed ledger. At this stage, each ledger (L1, L2, L3) is synchronized to ensure consistency across the Blockchain network. This distributed consensus mechanism guarantees data integrity and prevents malicious or invalid updates from being applied. To illustrate the practical operation of this framework, we consider a data offloading scenario. In this case, the device h1 can offload its data either directly via the OpenFlow switch connected to the SDN domain, or indirectly through device-to-device (D2D) communication with h3. This dual-path approach highlights the flexibility of the proposed architecture in managing different communication modes within small cells.

To evaluate the performance of the system, the offloaded data is segmented into flows of varying sizes, which are stored in the Blockchain. The considered data sizes include 134 bytes, 275 bytes, 301 bytes, and 1.875 kilobytes. These values demonstrate the ability of the Blockchain to handle heterogeneous data flows, ranging from lightweight IoT sensor data to larger multimedia packets. The results obtained from these experiments provide insight into the scalability and efficiency of our Blockchain-based data offloading mechanism.

Figure 2.5 illustrates the latency performance of the proposed architecture, which integrates the Hyperledger Fabric Blockchain for secure data offloading, evaluated as a function of simulation time. To configure the experimental environment, flow rules were defined in the SDN-based network through a REST API, enabling the creation of specific offloading paths between devices. This approach allows a precise control over data forwarding decisions and facilitates the monitoring of flow dynamics across the system. Once the network configuration was established, the latency results were collected using the following REST API endpoints:

- $Htttp : 127.0.0.1/stats/switches$ for retrieving information about the active switches in the topology.
- $Htttp : 127.0.0.1/stats/flow/ < dpid >$ for obtaining detailed flow statistics associated

with a given datapath identifier (DPID).

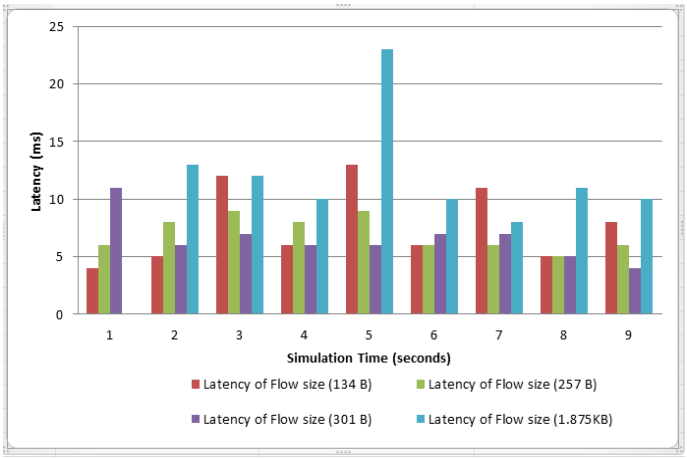


Figure 2.5: Latency performances of the proposed architecture.

These endpoints provide real-time insights into the operation of the network, making it possible to observe latency behavior under various offloading paths and flow conditions. The obtained results demonstrate that the latency performance remains stable across different paths and under multiple flow sizes. More specifically, the latency values observed are very similar, and there is no significant increase even when the simulation time is extended or the block size within the Blockchain grows. This consistency is a direct outcome of the lightweight integration of Blockchain mechanisms within the SDN-MEC environment. The proposed architecture achieves these results by minimizing Blockchain-related overhead while maintaining a secure offloading process. The distribution of control through SDN controllers combined with the use of Hyperledger Fabric enables efficient validation and block creation, without introducing excessive delays. As a result, the system not only preserves data integrity and security guarantees, but also ensures low-latency performance, which is crucial for IoT and mobile edge computing environments.

Conclusion

Although Edge Computing networks have attracted significant interest from both industry and academia in recent years, the design of secure mobile offloading architectures remains a considerable challenge. This difficulty primarily stems from the resource limitations of IoT devices, combined with the intensive computational workloads required by modern applications. Such constraints highlight the need for innovative solutions capable of ensuring both efficiency and security in dynamic edge environments.

To address this issue, we proposed a multilayer MEC architecture specifically designed to enhance security during the offloading process. The proposed framework ensures that mobile devices can securely offload their data to the central cloud when connected via mobile networks. Additionally, the architecture supports alternative offloading paths through Wi-Fi or D2D communications, leveraging a private blockchain network to guarantee integrity, confidentiality, and trust among participating devices.

In the evaluation phase, the proposed system was implemented and tested using a combination of the Mininet-WiFi emulator and the Hyperledger Fabric platform. These tools allowed us to emulate realistic IoT and MEC scenarios while assessing the security and performance of our solution. The results demonstrated that the architecture consistently provides stable security performance across multiple offloading paths and under varying flow sizes, while maintaining low latency and minimizing blockchain-related overhead.

Chapter 3

Intelligent optimization of computing task management

Chapitre 3

Introduction

The emergence of MEC has significantly transformed the landscape of distributed computing by providing substantial improvements in latency reduction and throughput enhancement for a wide range of applications and services. By deploying computational resources closer to end users, MEC enables real-time processing and supports the growing demand of latency-sensitive applications. However, the integration of streaming services within MEC frameworks remains particularly challenging. The main difficulty arises from security concerns, given the inherent computational and processing limitations of edge nodes. Ensuring secure and reliable streaming under these constraints is complex, as edge devices often lack the computing power required to perform advanced cryptographic operations or intrusion detection tasks without compromising real-time service delivery.

Addressing these challenges calls for distributed and flexible solutions, capable of balancing the need for strong security mechanisms with the stringent performance requirements of streaming applications. In this context, NFV has emerged as a promising paradigm. NFV decouples network functions from dedicated hardware, allowing them to be virtualized and executed on general-purpose infrastructures. By leveraging NFV, it becomes possible to dynamically allocate and orchestrate resources to perform security tasks at the network edge without overburdening constrained devices.

This chapter introduces a distributed management solution that integrates NFV capabilities to strengthen the security of streaming services while preserving high performance. The proposed approach relies on NFV Orchestrator (NFVO) nodes, which are responsible for coordinating the allocation of resources and the enforcement of security policies across MEC infrastructures. By distributing security functions across orchestrated virtual environments, our solution alleviates the pressure on individual edge nodes and ensures efficient handling of real-time streaming workloads.

The evaluation of this solution demonstrates its effectiveness in achieving an optimal trade-off between security enforcement and system performance. In particular, the NFVO-based allocation model significantly outperforms both priority-based and heuristic allocation models, by offering more intelligent resource distribution and better orchestration of virtualized functions. These results confirm that NFV-driven distributed management not only enhances security for streaming services but also provides a scalable and efficient strategy for MEC-based architectures operating under resource constraints.

3.1 Distributed Network Security Management

Figure 3.1 presents the proposed distributed MEC architecture designed to address overload control and security management challenges in mobile edge environments. The architecture is organized into three principal layers: the core layer, the control layer, and the radio access layer. Each of these layers plays a complementary role in ensuring the efficiency, reliability, and security of the system.

The core layer represents the uppermost part of the architecture and is primarily responsible for providing a global view of the network. It handles high-level management functions such as resource orchestration, service aggregation, and global policy enforcement. By centralizing strategic decisions at this level, the architecture ensures consistency across the entire system and facilitates coordination among distributed entities. The control layer, situated between the core and access layers, plays a more localized and operational role. It enables distributed management close to the edge network, reducing dependency on centralized entities and minimizing latency. This layer incorporates mechanisms such as distributed SDN controllers and VNFs, which contribute to both load balancing and security enforcement. Its distributed nature makes it particularly well-suited for managing real-time workloads in dynamic and resource-constrained MEC environments. The radio access layer forms the foundation of the architecture by establishing wireless connectivity between end-user devices and the network infrastructure. This layer encompasses various access technologies such as small cells, Wi-Fi links, and D2D communications, providing the fundamental connectivity required to offload and process data at the edge. Its role is crucial in ensuring seamless access while supporting diverse traffic patterns generated by mobile and IoT devices.

By combining the strengths of these three layers, the proposed architecture achieves an effective balance between global oversight, localized management, and end-user connectivity. This multilayer integration not only reduces the risks of network congestion and overload but also ensures that security mechanisms are enforced consistently across the system. As a result, the architecture provides a robust foundation for reliable, secure, and optimized service delivery in mobile edge computing environments.

3.1.1 Core layer

The core layer provides overall management of the network and available resources through a set of strategic decisions regarding resource allocation, security policy management and network planning. It typically hosts network management applications, monitoring and control systems, and orchestration functions to manage task distribution and load balancing. Within this layer, the orchestration mechanism is responsible for network functions virtualization (NFV) and dynamic deployment of security functions on Edge nodes in response to the changing needs of the security workload.

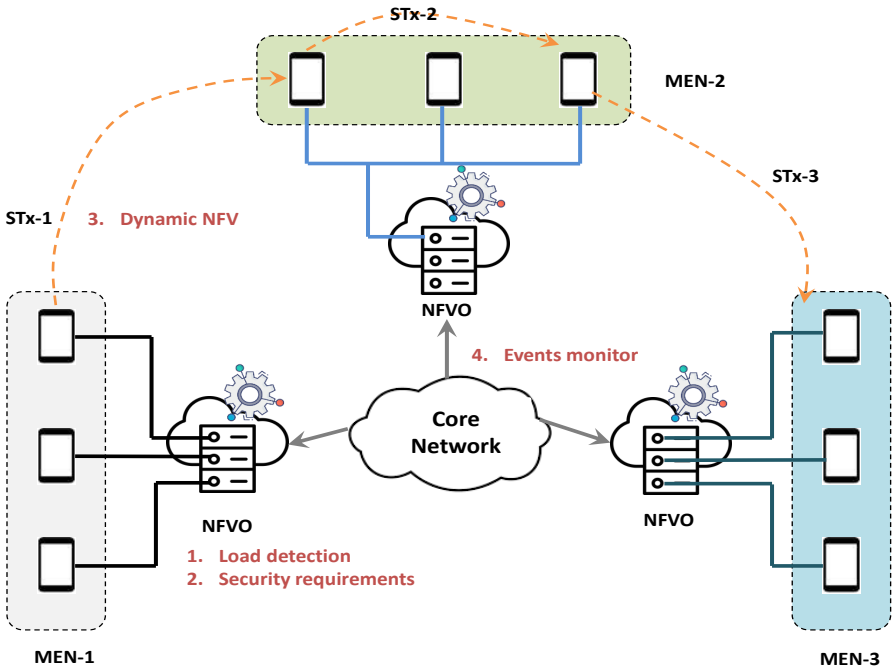


Figure 3.1: Distributed MEC architecture for overload control and security management.

3.1.2 Distributed Control layer

The distributed control layer is responsible for managing and controlling the resources of each Mobile Edge Network (MEN) as well as each Edge node locally. Its main function is to orchestrate operations (STx) in a given area, including deployment and management of applications, configuration of network equipment, management of local security policies, etc. It forms the link between the central layer and radio access by transmitting management and control instructions to network equipment and local levels.

3.1.3 Radio access layer

The radio access layer is the physical layer of the network that manages wireless communications between edge nodes (eNs) and Network Function virtualization Orchestrator (NFVO) nodes. This layer helps provide reliable, high-speed wireless connectivity to end users, using technologies such as Wi-Fi, cellular networks (LTE, 5G), etc. It also integrates mobility, quality of service (QoS) and security management functionalities at the wireless network level.

3.2 NFV security tasks

In this section, we formalize the notations and parameters used throughout this chapter (Table 3.1). These notations provide a precise framework to describe the requirements, allocation, and execution of security tasks within the proposed architecture. In particular, we highlight how computing, memory, and bandwidth resources are quantified and mapped to the demands of

secure streaming services.

Table 3.1: Notations.

Notation	Description
MEN	Mobile Edge Network
eN	Mobile edge node.
STx	Security task of a policy 'x'
D	represents secure streaming requests.
T_{stp}	The security task processing time
θ	Threshold.
$C(.)$	eN Capacity.
NFVO	Network Function Virtualization Orchestration
R_{cpu}	Available CPU resources
R_{mem}	Available Memory resources
R_{bw}	Available Bandwidth resources
P^i_{cpu}	CPU Resource requirements for security policy *i*
P^i_{mem}	Memory Resource requirements for security policy *i*
P^i_{bw}	Bandwidth Resource requirements for security policy *i*
A^i_{cpu}	CPU Resource allocations for security policy *i*
A^i_{mem}	Memory Resource allocations for security policy *i*
A^i_{bw}	Bandwidth Resource allocations for security policy *i*

3.2.1 NFV resources requirements

The resources available in a MEC environment are used dynamically and flexibly, depending on the processing needs of various NFV functions. The available computing capacity is determined by the number of virtual CPU cores available on each MEC processing node. It is essential to note that the sum of computing resources allocated to all security policies should not exceed the available resources, i.e: $A^i_{cpu} + A^{i+1}_{cpu} + ... \leq R_{cpu}$ of which each allocation must satisfy $A^i_{cpu} \geq P^i_{cpu}$ for each policy. The distributed execution of security policies in MEC nodes result in significant usage of available memory due to encryption/decryption operations, temporary storage of sensitive information, caching, etc. So, it is crucial to consider these memory requirements in the previous computing solution, i.e.: $A^i_{mem} + A^{i+1}_{mem} + ... \leq R_{mem}$ of which each allocation must satisfy $A^i_{mem} \geq P^i_{mem}$ for each policy. At the same way, the security needs previously mentioned can result in increased bandwidth usage. This use must respect the available bandwidth, i.e.: $A^i_{bw} + A^{i+1}_{bw} + ... \leq R_{bw}$ of which each allocation must satisfy

$A^i_{bw} \geq P^i_{bw}$ for each policy. By taking these measures, it is possible to ensure a secure streaming solution while guaranteeing a smooth user experience.

3.2.2 Resource Allocation Process For Security Policies in Edge Layer

The algorithm 1 describes the resource control, allocation and orchestration mechanism for streaming security overheads. This mechanism relies on the identification of streaming-related security congestion points, followed by their distribution to neighboring edge nodes available in the network. Edge nodes include access points, base stations and neighboring nodes. In order to maintain optimal performance of each edge node, we suggest fragmenting each security policy 'x' associated with streaming into a set of elementary operations, denoted by STx. Figure 1 illustrates the orchestration process between these nodes using NFVO nodes that implement network functions virtualization (NFV). By leveraging this virtualization and orchestration, it becomes possible to dynamically and efficiently deploy security features where they are needed most, based on the specific requirements of each streaming application.

Algorithm 1 NFVO Allocation Control for Resource Management and Security Policies
Input: D, STx

```
for each eN in NFVO control zone do
    O ← Overload(eN)
    if O > θ(O) then
        eNList ← IdentifyAvailableNodes(eN)
        for each OtherNode in eNList do
            if C(OtherNode) > Load(OtherNode) then
                DistributeTasks(O, OtherNode, STx)
                break
            end if
        end for
    end if
end for
```

To manage streaming-related security overheads, we present an Edge allocation process as a set of key steps:

Overload detection

Initially, the NFVO identifies areas where security requirements for streaming cause excessive resource utilization on Edge nodes. It then continually monitors the processing load of each policy on these nodes, along with the ongoing requirements of each policy. The arrival of secure streaming requests on each Edge node is modeled using a probabilistic distribution function defined as:

$$P(X = D) = \frac{e^{-\lambda}.\lambda^{D}}{D!} \tag{3.1}$$

Of which $P(X = k)$ represents the probability of D streaming requests arriving within a given time interval; λ denotes the average arrival rate of streaming requests per unit time; D stands for the number of streaming arrivals.

Next, an NFVO evaluates the performance and capabilities of Edge nodes to handle secure streaming requests. To do this, it uses a probabilistic estimate of the processing time for each security task, taking into account the average processing time μ and the standard deviation σ (which measures the dispersion of the values around the average) depending on the characteristics of the security task and the load of the Edge node. The security task processing time estimation function is defined as follows:

$$f(T_{stp}) = (\sqrt{2\pi\sigma^2})^{-1}.e^{-\frac{(T_{stp}-\mu)^2}{2\sigma^2}} \tag{3.2}$$

Security requirements

the requirements are analyzed to determine necessary security tasks, such as encryption, integrity verification, etc. In particular, an NFVO node calculates the total security load of each policy by multiplying the number of streaming requests by the average processing time for each task and then summing these products for all security tasks associated with streaming on the Edge node, that is:

$$S(D) = \sum_{i=1}^{n} d_i.f(T_{stp(i)}), d \in D \tag{3.3}$$

Dynamic NFV Allocation

the NFVO nodes continuously monitor the security load on each Edge node. When an Edge node exceeds its overload threshold, this condition is detected. Once the overload is detected, NFVOs evaluate the availability and capacity of other Edge nodes in the network. They identify available nodes that can support part of the security load of the overloaded node.

The NFV Orchestration and Event monitor

This orchestration mechanism with NFV enables agile and efficient management of security tasks between Edge nodes. First, NFVO monitors identify the real-time security requirements of each policy and collect information about the current security load, overall performance and capacity of Edge nodes in the network. Based on this information, it is possible to optimally use the available resources and ensure a rapid response to fluctuations in the security load. These decisions can be taken into account by the NFVO controllers by dividing each security policy $'x'$

associated with streaming into a set of elementary policies, STx. Then, each controller calculates the overall performance of each Edge node based on the elementary policies as follows:

$$P_{eN} = iP_{eN} - \alpha \left(\frac{N(ST_x)}{C_{eN}} \right) \tag{3.4}$$

Where P_{eN} represents the initial performance of the Edge node. α is a performance reduction coefficient depending on the number of STx operations; $N(ST_x)$ is the number of elementary operations associated with security policy $'x'$; C_{eN} is the processing capacity of the Edge node.

3.3 Performance evaluation

In this section, our study focuses on resource allocation in an MEC environment. We seek to evaluate the performance of our theoretical resource allocation approach for handling secure streaming requests in this environment. In particular, we evaluate the resource allocation performance of an NFVO under OMNeT++ [134]. The objective is to compare the performance of our model with two other allocation models: a model based on priority and another model based on a heuristic approach. For this comparison, we precisely extract performance using the following criteria:

- **Execution time:** We measure the time needed to execute the resource allocation for the security policies of each simulation model.
- **Resource Efficiency:** This metric evaluates the efficient use of available computing, storage, and communication resources. Methods that manage to maximize resource use while minimizing waste will be considered more efficient.
- **Scalability:** This metric evaluates the ability of each allocation model to adapt to varying workloads and an increase in the number of edge nodes. Methods that seamlessly adapt to changing streaming needs will be considered more effective.

The study scenario is simulated using the simulation parameters of the table 3.2, while defining resource management policies for each edge node.

Table 3.2: Simulation parameters.

Parameter	**Value**
Number of edge nodes	[5, 100]
Number of security policies	[5, 50]
Streaming requests	[5, 50]/s
Overload threshold	0.8
Qos threshold	0.9

The figure 3.2 shows the simulation performance for the three allocation models, comparing them based on the increasing number of security policies. Increasing this number also causes an increase in streaming security overhead, which impacts the performance of resource allocation models. Our results demonstrate that the NFVO allocation approach, implemented in our model, generally displays shorter execution times, thus demonstrating more efficient resource management compared to other models. In contrast, the priority-based allocation model generally exhibits higher execution times than the NFVO model, revealing lower efficiency in resource management, especially when the number of policies increases. On the other hand, the heuristic approach shows variable execution times, thus highlighting the sensitivity of this model to other specific conditions.

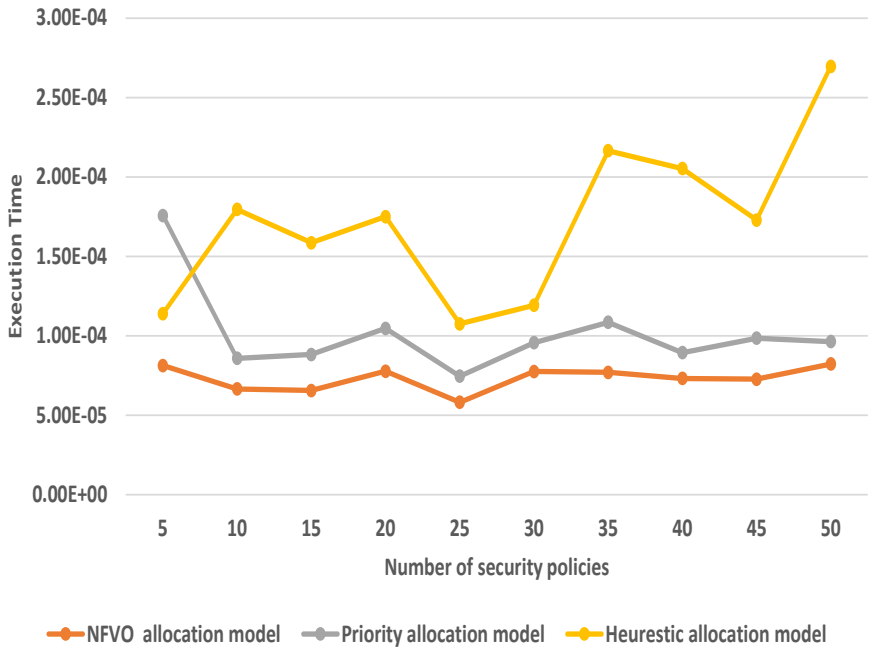


Figure 3.2: Execution time performance.

The figure 3.3 shows the scalability performance of the three allocation models, comparing them as the number of edge workers increases. Increasing this number also leads to an increase in the complexity of streaming security, which impacts the performance of resource allocation models. The performance shows that our proposed NFVO approach maintains relatively stable execution times even with an increase in the number of nodes, which ensures better scalability compared to other models. The priority-based model and heuristic approach show a significant increase in execution time with an increase in the number of nodes, indicating limited scalability.

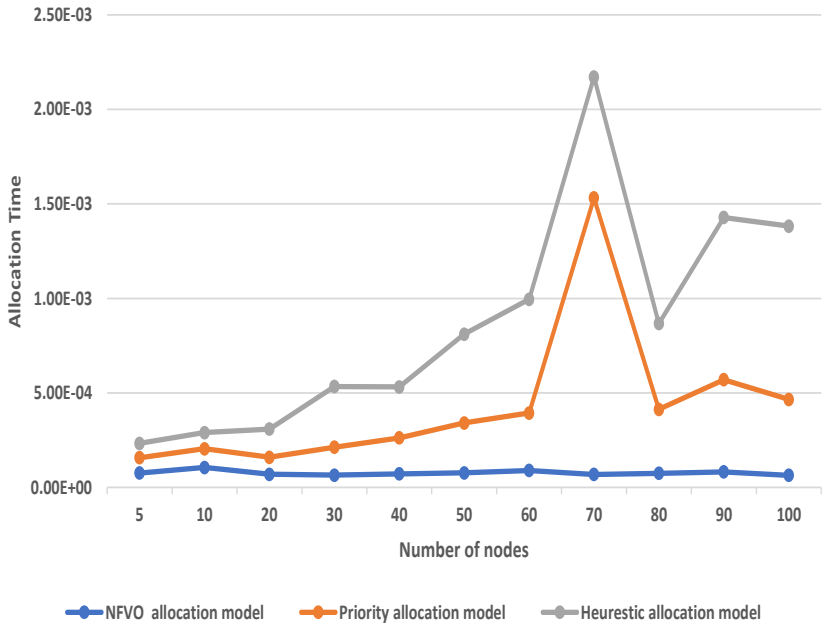


Figure 3.3: Scalability performances.

Chapitre 3

Conclusion

Our contribution lies within the broader objective of optimizing the security of streaming services in MEC environments. Given the inherent constraints of MEC infrastructures, particularly the limited computational and processing resources of edge nodes, traditional security solutions often prove insufficient to meet the dual requirements of real-time performance and robust protection. To address these challenges, we proposed a distributed management solution that harnesses the flexibility of NFV. By virtualizing security functions and orchestrating their execution across MEC nodes, our approach ensures that security tasks are dynamically allocated in accordance with the available resources and the requirements of ongoing streaming services.

The evaluation phase compared our NFVO-based resource allocation model with conventional allocation strategies, including heuristic and priority-based models. The results clearly demonstrated the superiority of the NFVO approach, particularly in terms of resource utilization efficiency, latency reduction, and stability under varying workloads. This confirms the potential of

our solution to significantly enhance streaming security, without compromising the overall performance and scalability of MEC systems. Looking forward, this work opens several promising research directions. Refining the orchestration algorithms, integrating machine learning techniques for predictive resource allocation, and extending the model to multi-domain MEC scenarios are natural continuations of this research. By advancing these perspectives, our proposal contributes to shaping the future of secure and efficient streaming services in next-generation edge computing environments.

Chapter 4

Optimizing network performance and security in video surveillance data storage

Chapitre 4

Introduction

Video surveillance systems have experienced exponential growth over the past decade, both in the public (urban security, critical infrastructure monitoring, transportation) and private (home automation, businesses, healthcare facilities) sectors. This expansion is largely driven by technological advances in high-resolution sensors, intelligent video analytics, and increasing interoperability with IP networks, making systems more flexible, connected, and integrable with cloud or edge solutions. However, this evolution is accompanied by new threats, particularly related to the sensitive and private nature of the captured data, their transmission over unsecured networks, and the risks of alteration or tampering.

In this context, the issue of video stream storage security is becoming a central concern. The massive transfer of video to cloud or remote servers not only poses bandwidth and latency issues, but also increases the potential attack surface. The compromise of a network link or server can thus expose critical data, harm individual privacy, or compromise the legal value of visual evidence. Faced with these challenges, the emergence of the MEC paradigm offers a relevant solution: by bringing processing and storage closer to data sources (cameras, local gateways), it is possible to reduce transmission costs, improve responsiveness, and better control local security.

At the same time, blockchain technologies introduce unprecedented guarantees in terms of data traceability, integrity, and authenticity. In particular, the use of smart contracts makes it possible to define automated and verifiable data management policies, while optimizations such as Segregated Witness (SegWit) or the use of consortium blockchains make it possible to maintain performance compatible with the real-time constraints of MEC. In addition, the IPFS represents a powerful alternative to centralized storage models: it allows videos to be distributed across different nodes while ensuring their integrity through hashing mechanisms, making the system more robust, decentralized, and resilient to attacks.

In this chapter, we present the combination between these technologies to meet the security, scalability, and performance requirements of modern video surveillance systems. We propose a hybrid architecture based on MEC, Blockchain, and IPFS, where each captured video is stored in a distributed manner, controlled by smart contracts, and verified in real time using digital fingerprinting mechanisms. We describe in detail the system design, its functional components, interaction protocols, and the experimental evaluation demonstrating its superiority in terms of reliability, latency, and security compared to existing approaches.

4.1 System overview

This section provides a comprehensive description of the VSS within a MEC environment. It highlights the fundamental requirements for secure data transmission, discusses the associated security challenges, and introduces the main notations that will be used later to formally express storage operations and integrity verification mechanisms. The objective is to clearly define the foundations of the system before presenting the detailed architecture and security protocols.

4.1.1 VSS data requirements

The proper functioning of a VSS in a MEC context depends on several critical requirements that must be satisfied to guarantee both performance and security. These requirements can be grouped into three major categories:

Security requirements

The private nature of video surveillance data requires a set of regulations when using the system (connection, access, authorization, etc.) and data protection guarantees (reliability, confidentiality, etc.). To achieve these goals, it is necessary to have a high-level, immutable, simplified, and effective network security model. Thus, the ability to process and verify system access attempts and prevent security threats caused by untrusted third parties.

Our VSS operates in a secure private network, ensuring that the cameras and central server are protected against unauthorized access from the outside. This network is made up of several smart IP cameras strategically placed to cover critical areas. These cameras are connected to a central server capable of storing and processing large amounts of video data. The server uses secure communication protocols (HTTPS / TLS) to receive video streams, which are then encrypted using appropriate encryption functions to ensure confidentiality during storage and transmission.

System efficiency

The integration of the blockchain in an environment with constrained resources (edge nodes) causes additional overhead on network performance, and even the system cannot achieve the desired level of security. The integration model must therefore be effective and transparent at edge nodes. In other words, all edge nodes in the network must participate in the consensus process and be well suited to MEC environments.

To ensure effective VSS security in a resource-constrained environment, it is crucial to efficiently manage the storage of the vast volumes of video data generated by surveillance cameras, while ensuring their security and availability. We consider several efficiency requirements, including video segmentation, distributed storage, and the use of blockchain. To this end, we use the InterPlanetary File System (IPFS), which distributes video data among multiple network

nodes. This approach helps reduce the load on each individual network node and ensures high redundancy as well as increased data availability.

Blockchain scalability

The VSS generates a massive volume of video data over time, requiring minimal storage latency. If each node on the blockchain stores the entire video segment, this can quickly saturate the processing resources available at the edge nodes and even lead to high latency. Therefore, blockchain scalability must include efficient storage management mechanisms to avoid security fees. In addition, the blockchain consensus and processing mechanisms must also be fast enough to meet these requirements.

To ensure blockchain scalability in VSS, we have integrated the Segregated Witness (SegWit) framework. The SegWit structure helps increase the capacity of the blockchain by separating transaction signature data from the rest of the transaction data. This separation allows transaction sizes to be reduced and more transactions per block to be included, thereby optimizing bandwidth usage and reducing processing load on nodes. Additionally, SegWit improves security by solving the problem of transaction malleability and facilitates the adoption of future scalability solutions. These improvements are crucial to maintaining high efficiency in resource-constrained environments, ensuring that our VSS can scale and adapt to growing needs without compromising performance or security.

4.1.2 Notations

The following table lists the basic notations used in this chapter to describe the theoretical approach of our proposals.

These notations make it possible to formalize the technical and security aspects of the system, thus facilitating the understanding and implementation of the solutions proposed for the storage and verification of the integrity of video data.

4.2 System architecture

The figure 4.1 illustrates the MEC architecture we propose for secure storage of VSS data. This architecture is composed of three fundamental elements: the VSS, a distributed edge layer, and the user layer. The data protection process begins when the video is generated by the surveillance system, continues with its storage in IPFS, and finally ends with its transfer through an SDN domain. It extends until the video is received and verified by authorized entities in the real world.

Table 4.1: Notations

Notations	Description
$Sg.$	Video segment
$M(.)$	Masking video data function
$E(.)$	Encryption video data function
$V(.)$	Verification function
T_s	Video Timestamp
M_d	video meta data
V_n	video version
$h_x(.)$	Hash function
Pk	Public key
Sk	Private key
$VCID$	Video Content IDentifier
B_i	i-th block of the BC
G	Generator on ECC
p	prime order
K_{1i}	The seed key of i-th video segment
N	The subgroup order of points on ECC curve

4.2.1 VSS

The recording process of the VSS involves the use of smart cameras linked to the video server. For the purposes of this study, we consider the VSS system to be part of a private network or secure computing environment. In particular, the video data captured by each camera is transmitted to the video server via a secure channel. The video server is provided with sufficient computing and storage resources to implement the necessary cryptography measures. During recording, video clips are divided into smaller segments, denoted $Sg(i)$, while preserving their chronological order on the server. Unlike the partial data hiding approach adopted by the architecture described in reference [135], our study assumes that the confidential data contained in the video segments is fully hidden using the function $M(Sg(i))$ and is encrypted on this server using the function $E(M(Sg(i))$.

4.2.2 Distributed Edge Layer

The architecture of the edge layer we propose aims to decentralize the processes of storing and transmitting surveillance video data. This decentralization is achieved through the use of a decentralized storage protocol based on IPFS and the verification of transmission integrity through a signature verification system. Using this storage protocol, the system can store and share video segments with various organizations. Each video segment is identified within the IPFS system by a Video Content Identifier ($VCID$), which corresponds to the hash value of the video segment. This approach ensures secure and efficient retrieval of video segments later in the network. The components of the edge layer consist of fully distributed nodes, including blockchain nodes, IPFS storage nodes, and SDN controllers. This distribution enables

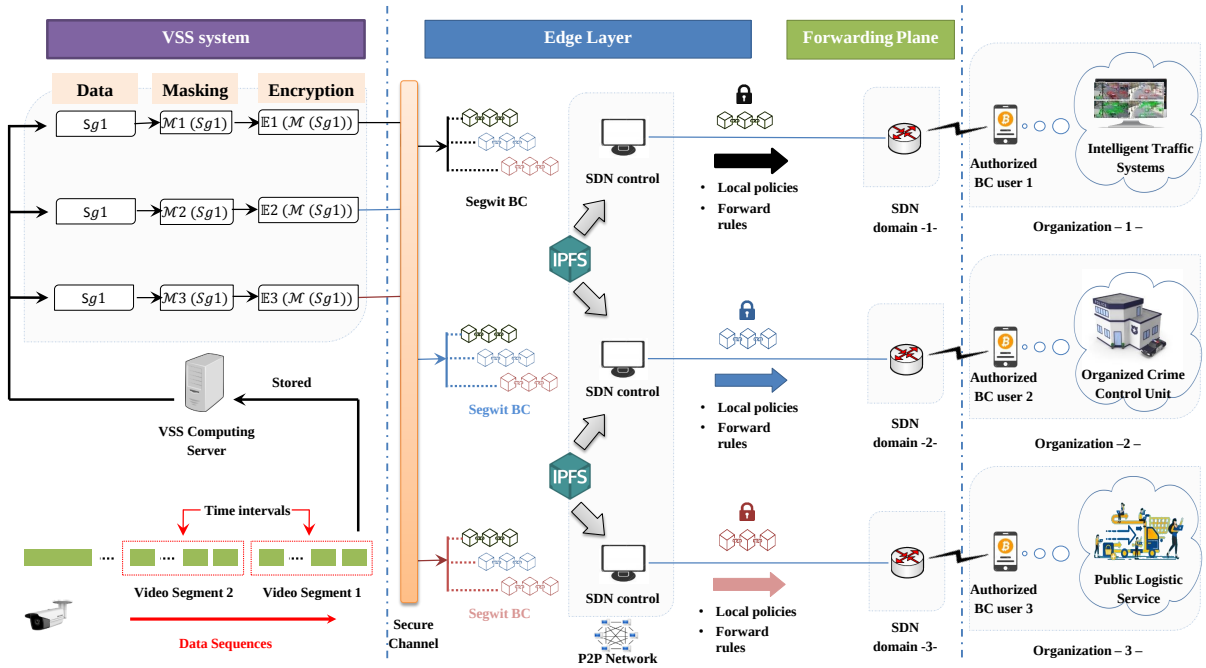


Figure 4.1: The distributed edge layer for the VSS system.

the implementation of a decentralized video data control system using blockchain structures while overcoming the security and scalability issues associated with centralized management. Furthermore, the edge layer can activate one or more types of blockchains ($BC(j)$) for each organization, offering maximum flexibility. To achieve these goals, we propose a simplified approach by separating the control processes via OpenFlow and integrity verification. In other words, surveillance video security is data-centric and managed by a decentralized SDN control plan. This decentralization approach can provide the following benefits:

- **The decentralized cross-domain of the edge layer**: This characteristic makes it possible to highlight the importance of inter-domain interaction in the proposed architecture. The aim of this feature is to ensure secure control of video surveillance data using a custom blockchain (BC) for each domain. This helps facilitate effective communication of essential information between these areas. This feature also helps reduce proof-of-work (PoW) requirements and load on the network, which is beneficial to overall system efficiency. Thus, to streamline the level of control by using smart contracts to automate the execution of shared agreements. This allows rapid validation of new transfer rules by all nodes in the blockchain network, which can then be transmitted to OpenFlow Switches (OFS). In other words, this feature aims to simplify and accelerate the management of data transfer rules within the network.

- **The decentralized intra-domain of the edge layer**: With this characteristic, the management and control of local policies for each organization are facilitated. This includes tasks such as verifying the connectivity of authorized nodes and monitoring

network performance when automatically routing video surveillance data between OpenFlow Switches (OFS) or applying new rules to these switches.

4.2.3 VSS user

The system users are organizations that can access the video surveillance service via authorized user data (AuD). These organizations can be traffic control units, physical crime control units, logistics services, etc. These organizations connect to the system using a set of system-assigned cryptography addresses, a key pair (public (Pk) and private (Sk)) to receive video data, and a symmetric key to decrypt downloaded video segments. The identification of each organization and the VCID of each segment are also used to access video content stored in IPFS. Once these organizations, through their mobile devices, have access to this network, they can connect and download the blocked content while verifying its integrity.

4.3 A SegWit blockchain model for Edge layer

In this section, we address specific challenges related to security and blockchain size scalability in the context of storing video segments in an MEC environment. These challenges include network overhead resulting from storing videos entirely on the blockchain as well as latency during video retrieval. We present a storage solution based on a scalable block structure while ensuring a high level of security. In particular, the solution is based on the use of Segregated Witness (SegWit) technology to optimize the efficiency of IPFS storage and data transmission. Specifically, SegWit integration improves security by allowing signature witnesses to be stored separately from master bloc data. This approach helps mitigate the risks of video tampering or transaction falsification. It also highlights video transaction storage policies and introduces additional security features, such as verifying the proof of integrity of videos stored in IPFS and validating the authenticity of associated transactions. This proof-of-integrity check provides a balance between transaction speed, security effectiveness, and network performance. Figure 4.2 illustrates the SegWit blockchain framework for storing and verifying the proof of integrity of video segments stored in IPFS.

The integrity proof of video data stored in IPFS serves to ensure the authenticity and integrity of these segments. It accomplishes this by verifying that the data remains unchanged and uncorrupted since its initial recording by smart cameras. Additionally, it guarantees immutability, protects the video from tampering, maintains traceability, and enables video auditability. This, in turn, strengthens trust in the data exchanged with organizations and ensures its long-term integrity. The mechanism for verifying the integrity proof relies on the integration of Segregated Witness (SegWit) structures to upgrade the capabilities of the proposed system, all without compromising network performance and efficiency. This mechanism takes into account the following specifics when using SegWit structures for video data security in the

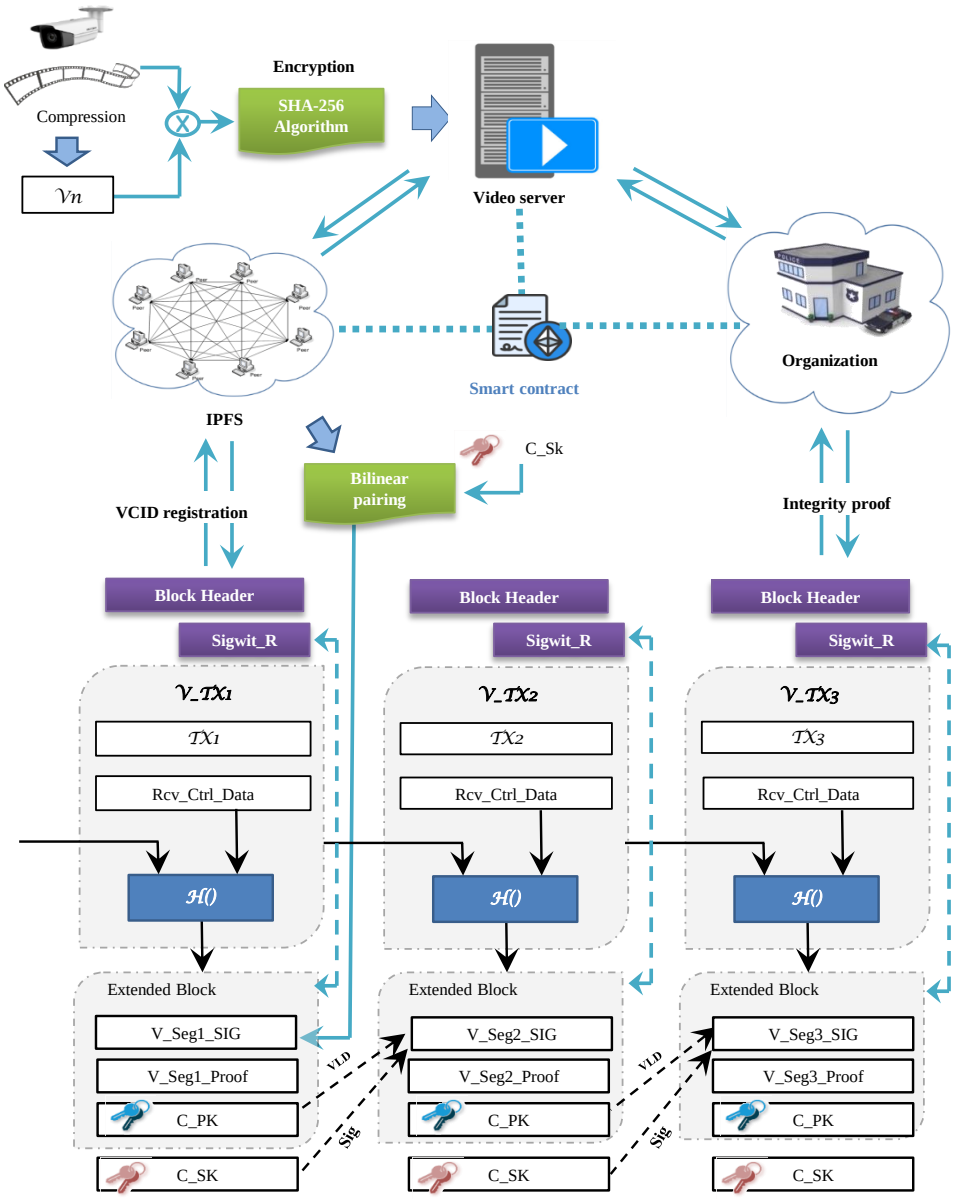


Figure 4.2: Illustration of a SegWit blockchain structure for video storage in IPFS and their proof of integrity.

edge layer:

4.3.1 ECC Key generation

The generation of public and private key pairs for VSS relies on the use of an efficient asymmetric elliptic curve (EC) cryptography system. This module uses elliptic operations to generate public keys on the curve, using random points that correspond to the private keys and the curve generator (G). The use of these cryptography systems offers significant efficiency advantages over the traditional RSA-based cryptography model. Additionally, these systems are widely adopted in applications such as cryptocurrencies, where they provide effective levels of security through reduced key sizes and faster key and signature generation speeds.

4.3.2 Video signature

The security of video data within VSS is preserved thanks to the signature mechanism, which creates a unique signature for each video segment using the SHA-256 hashing algorithm. To ensure a high level of protection, we employ a bilinear elliptic curve to generate short signatures efficiently, allowing video integrity to be verified. This cryptography approach offers effective detection of any attempt to manipulate or falsify the content of these segments. Indeed, any alteration of a segment would result in the rejection of its signature, thus reinforcing the guarantee of integrity and confidentiality of video content in the VSS. As mentioned earlier, we can ensure a trade-off between transaction speed, security effectiveness, and network performance by storing signatures in a block structure separate from the video transaction block. This block is then used by the integrity verification process to recover signatures and video evidence (see Figure 4.2). Furthermore, the signature of the entire block serves as a header for the next block.

4.3.3 Smart contract for record integrity proof

The proposed system operates in a decentralized manner through a set of smart contracts, thus ensuring the immutability of storage, reducing vulnerability to attacks, and strengthening the reliability of video integrity proofs. By deploying these smart contracts, the edge layer can guarantee the integrity of video recordings stored on IPFS as well as their integrity proof. In addition, it manages the identities and interactions between the nodes of this layer. All this simplifies the search, recovery, and subsequent analysis of video recordings. Additionally, the validity of video segments originating from IPFS nodes can be verified through the deployment of these smart contracts, providing integrity proofs regarding the following functionalities:

- Establish a distributed storage system for $M_d(S_q), i \in [1..n]$ metadata related to surveillance videos, using the IPFS protocol. This metadata includes essential information such as

time, date, location, and camera details. Through the use of blockchain, this metadata is recorded in a decentralized manner, which ensures its integrity and simplifies its subsequent retrieval. The smart contract makes it possible to establish proof of integrity on the data because each video segment is associated with a unique hash, $VCID = h(S_{qi})$, which can be verified at any time to ensure that no tampering has occurred.

- The smart contract is capable of automatically initiating the process of verifying the authenticity and integrity of video segments stored in IPFS. Specifically, the system can verify received video segments to ensure that they come from cameras authenticated by approved manufacturers. Only video transactions signed and transmitted by these cameras via a secure channel are considered legitimate, while video transactions from unauthorized sources are rejected. This measurement makes it possible to detect any alteration of the content of video segments or the use of non-certified material.

- Ensure the integrity of the IPFS storage through a set of system access control mechanisms. This ensures that only authorized organizations can access the IPFS system and video recording via a blockchain-validated hash. The organization can then verify the integrity of the recovered video by recalculating its hash and comparing it with the hash stored in the blockchain, as well as via the security measures linked to their proof. If the hashes match and the proof is validated, this confirms that the video data has not been modified since it was originally recorded.

The algorithm 1 describes the validation of video segments via a smart contract, the objective is to guarantee the authenticity and integrity of video segments in the VSS using blockchain and robust security protocols. While figure 4.3 presents the smart contract structure for received video streams validation. The algorithm takes video streams and their metadata as input and then determines whether the video segments are valid or not. First, the algorithm checks whether an edge node has received video segments via a secure channel. This initial check is crucial to ensure that video data has not been compromised in transit. If the video segments come from a secure channel, the algorithm verifies their authenticity and integrity using the HMAC-MD5 algorithm. HMAC-MD5 combines a secret key with the video message and applies an MD5 hash function to produce a hash value, which is then used to verify that the video segments have not been altered and come from an authentic source. If the authenticity and integrity of the video segments are validated by HMAC-MD5, the video data is sent to the DPoS (Delegated Proof of Stake) nodes. These DPoS nodes are responsible for validating transactions and maintaining consensus in the blockchain network. After sending to DPoS nodes, the algorithm returns ”Valid” status, indicating that the video segments are authentic and intact. On the other hand, if the authenticity and integrity of the video segments are not validated, or if the video segments were not received via a secure channel, the algorithm returns ”Denied”. This process ensures that only reliable and secure video segments are accepted into the system, thereby strengthening the security and integrity of data in the VSS.

Algorithm 2 Smart contract for video segment validation

Require: Video streams , $M_d(S_{qi}), i \in [1..n]$
Ensure: Valid , Denied
1: **if** an edge node received video segments from a secure channel **then**
2: Verify the authenticity and integrity of video segments using the HMAC-MD5 algorithm
3: **if** the authenticity and integrity are valid **then**
4: Send video data to DPoS nodes
5: **return** Valid
6: **else**
7: **return** Denied
8: **end if**
9: **else**
10: **return** Denied
11: **end if**

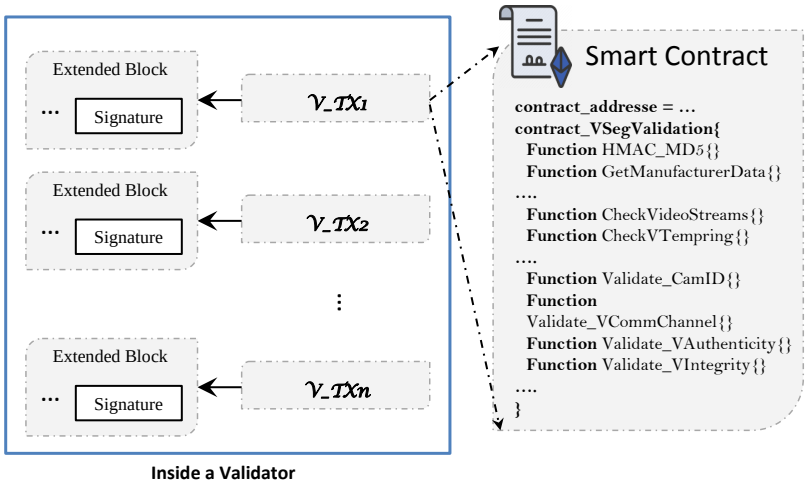


Figure 4.3: Smart contract for received video streams validation.

4.3.4 SegWit-based IPFS for Distributed Video storage

As we mentioned previously, the SegWit block structure design provides a scalable solution for decentralized IPFS storage on each edge node. Expanding the structure of each block allows for a larger number of VCIDs while keeping security information separate. This also translates into cost savings related to video transaction verification and validation operations. Video data is stored in IPFS using the following steps:

Generate a seed symmetric encryption key :

The video server initializes the decentralized storage process with the IPFS nodes and the blockchain nodes. First, the video server encrypts the received video segments with a symmetric encryption key K_{1i} generated by Elliptic Curve Diffie-Hellman ($ECDH$) key exchange. The video server initializes a starting symmetric key $K_{1(0)}$ with the organization $InitECDH(p, d_s, d_o, G, N, h) \longmapsto K_{1(0)}$, where d_s, d_o are random nonces $\in [1, N-1]$. The key $K_{1i}, i \neq 0$ is renewed regularly for each video segment according to the following equation:

$$K_{1i} = \begin{cases} D_{s,x}Q_{o,x} = D_{o,x}Q_{s,x}, & i = 0 \\ K_{1(i-1)} \oplus h(V_n + T_{root}), & i > 0 \end{cases} \tag{4.1}$$

It is possible to renew the symmetric key by adding the version of the current video segment V_n and the value of the previous encryption key $K_{1(i-1)}$ and the merkle root T_{root} of the sequence. Once the video server encrypts the video segment, it is transferred over the IPFS network. At this point, the file is subjected to a SHA-256 hashing operation by IPFS, ensuring that a unique and invariable identifier, the $VCID$, is assigned to each video segment. This is possible because the same video will always produce the same hash. The $VCID$ functions as a permanent identifier of video content on the IPFS network, providing a decentralized means of referencing and accessing the video segment. Saving the $VCID$ in a valid SegWit block ensures the integrity of video content stored in IPFS. If any alteration is made to the video, it will result in the creation of a completely different $VCID$. Therefore, if a third party attempts to manipulate the video segment, this manipulation will be detectable thanks to the change in the $VCID$.

In the decryption step, the organization can simply generate the seed symmetric key $K_{1(0)}$ using only its private key d_o and the public key $Q_{s,x}$ of the video server, because $d_oQ_{s,x} = d_od_sG = d_sQ_{o,x} = d_sd_oG$. Then, it is possible to calculate the symmetric key $K_{1i}, i \neq 0$ of each block by extracting their version V_n and the root merkle T_{root} of the received sequence. This type of storage involves robust and high-level security measures to protect video data, thus preventing unauthorized access and ensuring that the data remains intact and unaltered.

SegWit Blockchain :

To improve the scalability of the VSS security system while extending its capabilities while maintaining security, we designed a SegWit structure to store the video hash efficiently. This SegWit structure consists of four essential elements: the block header contains essential information (block number, timestamp, merkle root, etc.), a master block (transaction data), an extended block (transaction witness), and a witness of the merkle root calculated based on the witness data. The master block groups the main video transactions (TX), which include the VCIDs generated by the IPFS system, playing a crucial role in the construction of these transactions. Each transaction TX contains the hash resulting from the concatenation of a VCID, video metadata $M_d(S_{qi})$, as well as the hash of the previous block $h(B_{i-1})$, as shown in Equation 4.2. These transactions are then organized within the master block. The master block is linked to its extension block, the witness block, which stores verification information from edge layer nodes in the form of transaction witnesses. These witness transactions include signatures, scripts, and integrity proofs, essential for validating and verifying TX transactions. The block validation process is decentralized, involving edge layer nodes, ensuring the legitimate origin of video segments transmitted over the network. This structure strengthens security while optimizing video hash management, thereby improving overall VSS system performance. In the validation stage, a smart contract is used to verify the confidentiality and integrity of the recording of the received video segment, as well as when verifying the transmission of video

segments between organizations and the IPFS system.

$$TX_i = h\left(M_d\left(S_{qi}\right)||R_m\left(VCID_i\right)||h\left(B_{i-1}\right)\right) \tag{4.2}$$

In the validation stage, a validator begins examining hashes that are able to accurately satisfy a predefined difficulty. It only transmits valid hashes to other edge nodes connected to separate organizations. Rewards are granted only to the first validators who manage to resolve this difficulty. In this process, transaction data is consolidated into a master block, while witness data is consolidated into an extended block.

Then, the new block is broadcast to other validators to complete the validation process, which itself is reinforced by the two-step proof of integrity. The first step is to verify the correspondence between the master block data and its witness data block (signature, script, and integrity proof) to verify the validity of the SegWit block. The second step is to verify the match between the hashes of the data stored in the blockchain and the witness data. The proof of integrity verification confirms this validation through IPFS storage hashes and distributed video content routing, which acts as a mapping mechanism or an associative array of buckets. A block is valid if and only if the hash codes in the list of transaction data pointers match the test data in the extended block. This type of routing allows the edge node to map the master block to a network node that records the corresponding extended block.

Once the block data is validated, the blockchain nodes begin verifying witness transactions and transaction data to ensure the integrity and legitimacy of video transactions. At the beginning, the video transactions are retrieved from the master block, and the VCID is also retrieved from IPFS to extract the video metadata and associated test data. Then, the witness data is retrieved from the extension block to validate the test data using its metadata, signatures, and proof of integrity. If the test data matches the witness data, this confirms the integrity of the video storage and the legitimacy of video transactions in the system.

4.4 Video integrity proof verification scheme

In this section, we introduce a mechanism for validating the integrity proof of video storage within the edge layer. Our scheme encompasses a video server, IPFS nodes, blockchain nodes, and an organization. It makes it possible to verify the proof of integrity for both the storage and transmission phases. During the storage phase, the server encrypts the video segments and sends them to IPFS with the accompanying signature proofs. Subsequently, IPFS verifies these proofs and securely records the transaction within the master nodes through the use of a smart contract. In the transmission phase, an organization engages with the edge layer via a smart contract to initiate an upload process. Following this initiation, an organization generates a challenge to confirm the integrity of segments stored in IPFS. In response to this request, IPFS generates a proof of integrity, denoted P, corresponding to the challenge information. It then validates the

IPFS storage by storing the proof P as a transaction witness within the SegWit structure. This schema provides a simple verification step that an organization can easily implement to validate received segments. The verification details for each step are summarized as follows:

4.4.1 Secure video storage

Setup ($1^k \longmapsto (Pk, Sk)$) : Consider $G1$ and $G2$ as multiplicative cyclic groups with a prime number p, where g is a generator of $G1$. Additionally, let $f : G1 * G1 \longmapsto G2$ represent a bilinear mapping function, $h1 : \{0,1\} \longmapsto G$ is a hash function that converts a plaintext message into a point on the elliptic curve, $h2 : G \longmapsto Z_p$ is another hash function, $u \in G$, $Sk \in Z_p$ is the private key, and $Pk = g^{Sk}$ is the public key.

GenVtag $(Sk_{vs}, V_s) \longmapsto (NV_{ID}, T_s, R_{Tag}, \varphi, \sigma_G)$: This algorithm takes the video server private key Sk and the video sequence V_s as inputs. Initially, the video server divides the video V_s into m sequences $V_s = \{V_1, V_2, ..., V_m\}$. Then, it calculates a set of partial signatures $\varphi = \{\delta_{V_1}, \delta_{V_2}, ..., \delta_{V_m}\}$, where δ_i is a partial signature and calculated for each video segment $V_i, i \in [0, m]$ according to the following equation:

$$\delta_{V_i} = \left(h_1\left(VCID_i || T_{s(i)}\right) u^{V_i}\right)^{Sk}, i \in [0, m] \tag{4.3}$$

Additionally, the video server merges the video timestamp T_s and the organization identity (ID_{org}) to create a Network Video Identifier (NV_{ID}), where $NV_{ID} = h\left(T_s || ID_{org}\right)$ is a unique video identifier in the network. The $GenVtag$ algorithm is aimed to generate a Merkle root (R_{Tag}) for the video sequence V_s using the Network Video Identifier (NV_{ID}), as shown in the figure 4.4.

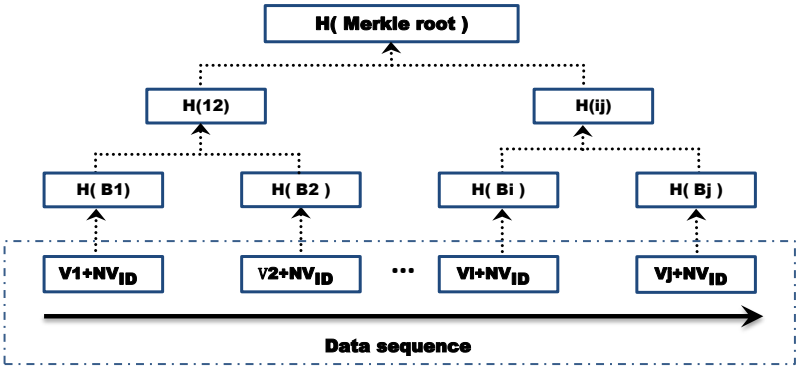


Figure 4.4: The Merkle root of V_s.

The leaves of the merkle tree represent the hash values of each block, of which each block merges the hash of the video segment V_i and the VN_{ID} using $h(B_i) = h(V_i || NV_{ID})$. To preserve both the integrity and the security of the parameters generated in this step, the video server sends (φ, NV_{ID}, T_s, R_{Tag}) to the smart contract for distributed storage

and $(V_i, T_s, \sigma_{sk}(V_i, T_s))$ to the IPFS system, where $\sigma_{sk}(V_i, T_s)$ is a proof signature of the corresponding message signed with its private key.

IPFS storage: Once the IPFS master nodes receive the triple $(V_i, T_s, \sigma_{sk}(V_i, T_s))$ from the video server, they perform signature verification with the video server public key to ensure the integrity of the received data. If the signature is valid, they send a registration request including the triplet $(NV_{ID}, ID_{IPFS}, T_s)$ to the smart contract to start safe video storage. Then, the smart contract responds to the IPFS master nodes with a registration identification (ID_{reg}). As soon as an ID_{reg} is received, the master nodes perform IPFS storage by calculating the $VCID$ of each $Vi \in V_s$. Next, they create a distributed IPFS mapping table to hold the video segments along with their $VCIDs$ and the IPFS nodes where they are stored, as shown in Table 4.2. Finally, each VCID will be sent with its ID_{reg} to the smart contract.

Table 4.2: The IPFS mapping table.

V_i	VCID	IPFS Nodes
{V1}	b56bed...ffb7d2212723	{1,3,7}
{V2}	edba9f...04273161968e	{2,5}
{...}	...	{...}

4.4.2 Video transmission

As part of transmitting video data to an organization with integrity verification, the process begins with the creation of a mobile wallet for the organization to receive the video streams and ensure their security. To achieve this, the organization must engage in a smart contract. In the proposed scenario, the interaction of the organization with the smart contract allows it to obtain an identifier ID_{org} and blockchain address (i.e., a public key Pk_{org} and a private key Sk_{org}). Then, it determines the Network Video Identifier $NV_{ID} = (ID_{org}||T_{s(i)})$ and sends it to the smart contract to retrieve the identifier of their registration ID_{reg} and all the $VCID$ identifiers. Once the organization receives the set of $VCID$ corresponding to the video timestamps T_s, it sends a verification task to the IPFS system. This system identifies the master nodes associated with the received $VCID_{(s)}$ according to the distributed IPFS mapping table. The verification steps are ensured by the following algorithms:

ChallengeGen ($K_{1(0)} \longmapsto (C, \sigma_C)$) : The goal of this algorithm is to establish and maintain a valid and verifiable proof of integrity for video data stored in both IPFS and the SegWit blockchain. In other words, this proof of integrity aims to reliably demonstrate that the challenged video segment blocks $1 \leq c \leq n$ are immutable and correctly managed, while preserving the confidentiality of the hash codes used to ensure integrity. It also certifies the secure storage and proper management of video data, thus reinforcing confidence in the

authenticity and integrity of these segments within the decentralized network. Based on the seed key $K_{1(0)}$, the organization generates a challenge request $C = (c, k1, k2)$, where $k1, k2 \in Z_p^*$ are two random keys to verify and ensure the block integrity during the verification process. Then, the organization sends the signed challenge request $\sigma_C = Sig_{sk}(C)$ with its private key to the IPFS system.

ProofGen ($(Q_k, S_q, C, \sigma_C) \longmapsto P$) : Once the IPFS system receives an access request redirected by the video server with the seed key $K_{1(0)}$, it confirms the received challenge using its signature and generates a subset of challenges $Y = \{c'_j\}$, where $j \in [1, c]$. Then, it calculates $C'_j = \pi_{k1}(j)$ and $V_{C'_j} = f_{k2}(j)$ for each $j \in [1, c]$ by PRP (pseudorandom permutation) and PRF (pseudorandom function), respectively. For each challenged video segment $j \in [1, c]$, the IPFS generates the proof parameters using the following equation:

$$P(C, u', \alpha, u, r) = \begin{cases} u' = \sum_{i \in (1,c)} V_i.VCID_i.g \\ \alpha = h_2(u') \\ u = u' + r.h_2(u^r), u^r \in G \\ (u, r) \in Z_p \end{cases} \tag{4.4}$$

Then, the IPFS calculates $\sigma_P = Sig_d(P||R_{Tag}(V_s))$ using its private key and broadcasts the couple $(\sigma_P, R_{Tag}(V_s))$ in the network.

Storage Validation: Once the pair $(\sigma_P, R_{Tag}(V_s))$ has been received by the blockchain nodes, a DPoS consensus is triggered to create a SegWit block. At first, the nodes verify the pair signature $V(Pk, \sigma_P, R_{Tag}(V_s)) = 1$ to confirm their integrity. If the integrity is verified, the blockchain nodes will trigger consensus with $Cb_s = (h(B_{i-1}), Mb, Eb, \sigma_s)$, where $h(B_{i-1})$ is the hash value of the previous block, Mb is the master SegWit block, Eb is the extended SegWit block, and σ_s is the signature of $(h(B_{i-1}), Mb, Eb)$. The delegators compete to ensure the following rule: "*no entity in common can control the security of the video traffic transmitted to the organization without proof (P). This proof is considered a puzzle game in which the execution of a correct integrity scheme presents a safer system and graded difficulties against the vulnerable aspects related to the video data.*". Based on this rule, a consensus response Cb'_s = (Mb', Eb', S_R) is generated and broadcast between the DPoS nodes, where S_R is the SegWit root of the new block. Each node receives Cb'_s, it verifies their integrity V(Pk, Mb', Eb', S_R) = 1 and the validity of the SegWit block (i.e., the correspondence between Mb' and Eb') using the following equation:

$$Succ(Mb', Eb') = \begin{cases} 1, Satisfy \\ 0, Otherwise \end{cases} \tag{4.5}$$

The success of the previous correspondence must satisfy the following conditions:

1. $Eb' = h(Mb'_{(i-1)}||VCID_i||P_i)$

2. $S_R = h(Eb'||\sigma_P)$

When a node within a blockchain receives multiple valid consensus responses, it incorporates the new block into its distributed ledger.

Proof checking: once the new SegWit block is validated, the IPFS master nodes request from the smart contract all partial signatures (i.e. φ) of the requested video segment. Then, they use $C'_j = \pi_{k1}(j)$ and $V_{C'_j} = f_{k2}(j)$ for each $j \in [1, c]$ to verify:

$$e\left(\delta_c^{\alpha}, g\right) = e\left(h_1\left(VCID||T_{s(c)}\right).u^{V_c \cdot \alpha}, g\right).e\left(u^{V_c}, Pk\right) \tag{4.6}$$

If the equation is valid, then integrity is proven, otherwise integrity is rejected.

4.5 Security Analysis

4.5.1 Correctness

Considering the security measures (δ, α, u) and the successful security of the proposed system, the organization and the IPFS master nodes can guarantee the integrity of the video data stored in the IPFS system. In other words, *only video segments stored in honest IPFS nodes can always pass the integrity verification process*. In this context, the correctness of this integrity check can be proved as follows:

$$\begin{aligned}
&e\left(h_1\left(VCID||T_{s(c)}\right).u^{V_c \cdot \alpha}, g\right).e\left(u^{V_c}, Pk\right) = \\
&e\left(h_1\left(VCID||T_{s(c)}\right).u^{V_c \cdot \alpha}, g\right).e\left(u^{V_c}, g^{sk}\right) = \\
&e\left(h_1\left(VCID||T_{s(c)}\right).u^{\alpha}, g\right)^{V_c}.e\left(u, g^{sk}\right)^{V_c} = \\
&\left(e\left(h_1\left(VCID||T_{s(c)}\right).u^{\alpha}, g\right).e\left(u, g\right)^{sk}\right)^{V_c} = \\
&\left(e\left(h_1\left(VCID||T_{s(c)}\right).u^{\alpha}, g\right)^{sk}.e\left(u, g\right)\right)^{V_c} =
\end{aligned}$$

$$\begin{aligned}
&\left(\left(e\left(h_1\left(VCID||T_{s(c)}\right).u, g\right)^{\alpha}\right)^{sk}.e\left(u, g\right)\right)^{V_c} = \\
&\left(\left(e\left(h_1\left(VCID||T_{s(c)}\right).u, g\right)^{sk}\right)^{\alpha}.e\left(u, g\right)\right)^{V_c} = \\
&\left(\left(e\left(h_1\left(VCID||T_{s(c)}\right).u, g\right)^{sk}\right)^{\alpha}\right)^{V_c} = \\
&e\left(\left(\left(h_1\left(VCID||T_{s(c)}\right).u^{V_c}\right)^{sk}\right)^{\alpha}, g\right) = e\left(\delta_c^{\alpha}, g\right)
\end{aligned}$$

4.5.2 Unforgeability

In the following, we will highlight the resilience of the proposed VSS against potential attacks, as well as the tamper-proofness of the results and the confidence in the validity of the storage integrity proof. We consider the following cases:

Case 1: A potential malicious storage script has the ability to analyze vulnerabilities in storage performed at the edge layer, and can potentially participate in attacks such as data alteration, substitution, or deletion. More specifically, this script could exploit sensitive user information, data stored in IPFS, the SegWit blockchain, as well as contested video blocks, in order to carry out a double attack: (a) The script is used to carry out a content forgery attack, which means deliberately altering stored data to mislead users or the systems that use it. (b) The script may interfere with the verification process, which could compromise the systems ability to ensure data integrity.

Validation results (case 1): the mathematical features of the encryption operations of the proposed VSS, such as bilinear mapping, signatures and hash functions, demonstrate that tampering with the security parameters would result in a detectable inconsistency. This also confirms the inability to tamper with integrity check results. Suppose that with this script, an attempts to modify one of the parameters of the following signature:

$$\delta_{V_c} = \left(h_1\left(VCID_c||T_{s(c)}\right)u^{V_c}\right)^{Sk} \tag{4.7}$$

Let $\delta^*_{V'_c}$ be the attempt to alter the signature of a contested video segment by the malicious script, where $V'_c \neq V_c$. The validation by the equation 4.6 will fail because the left side of the validation equation will be different from the right side, that is to say:

$$e\left(\delta^{*\alpha}_{V'_c}, g\right) \neq e\left(\delta^{\alpha}_{V_c}, g\right) \neq e\left(g, g\right) \tag{4.8}$$

However, $VCID_c$ and $VCID_{c'}$ do not share the same hash $h(VCID_c) \neq h(VCID_{c'})$ due to the collision resistance of the hash function. Therefore, the signature $\delta^{*\alpha}_{V'_c}$ does not match $h(VCID_c)$, leading to another verification failure. Additionally, assume that an attempt to substitute the public key (Pk') in place of Pk applied in the bilinear coupling as follows:

$$e\left(\delta^{\alpha}_{c}, g\right) \neq e\left(h_1\left(VCID||T_{s(c)}\right).u^{V_c.\alpha}, g\right).e\left(u^{V_c}, Pk'\right) \tag{4.9}$$

Therefore, the bilinear property of the coupling will detect this substitution, because the two sides of the equation will not be equal. Let also assume that the script successfully concealed information related to the validation process. The script cannot verify such a property without knowledge of the random variables $(u, r) \in Z_p$, which also results in validation failure according to equation 4.6.

Case 2: An independent verification attempt by malicious script could aim to manipulate or deceive the transaction validation process. This malicious attempt could involve including

incorrect data or manipulating signature scripts in a transaction, with the aim of fooling network nodes during validation.

Validation results (case 2): The SegWit block separates the data signature from the video segment, allowing independent verification. In case of malicious script injection, the validity of the signature still depends on whether it matches the original video transactions. Additionally, video data verification can be done separately. This separation allows verification of the block and transaction header without requiring the full download of signature data. This verification autonomy improves the efficiency of system security as well as the speed of the verification process. Therefore, even if there is a malicious script injected into the signature video data, authorized SegWit nodes will not download it, thus preserving the integrity of the verification process.

4.6 Evaluation and Simulation Results

The performance evaluation of our system is based on several key metrics, chosen to reflect the effectiveness of our approach in terms of security, scalability, and latency. We measured performance in a simulated environment under Omnet++, configured with a central video server, multiple MEC nodes, and end-clients accessing videos stored on IPFS.

- ***Storage and transmission latency:*** These metric measures the time that has elapsed between the time that a video is saved on the server and its availability through the MEC network. We conducted tests on various video sizes (from 2 MB to 100 MB) and compared the observed latency with traditional cloud-based storage solutions and distributed systems such as SWARM and Hyperledger Fabric.

- ***Optimized storage capacity:*** We measured the storage space used by our approach compared to traditional solutions by analyzing the impact of the SegWit blockchain on reducing storage load. The comparison was made by evaluating the size of the blocks generated for different scenarios of videos stored on IPFS.

- ***Transaction rate per second (TPS):*** This metric evaluates the ability of the system to validate and store video transactions on the blockchain. We compared our SegWit approach with solutions based on ECDSA-P256 and RSA-4096.

- ***Video integrity verification:*** To ensure that videos stored on IPFS have not been tampered with, we measured the validation time for integrity proofs. We compared our SegWit-based approach with methods using Merkle Tree and RSA, measuring the time required to verify the authenticity of 50 video segments.

- ***Bandwidth impact:*** To measure the efficiency of video transmission via MEC, we evaluated bandwidth consumption and the impact of video segmentation. We used

a network throughput of 100 Mbps and observed that our approach reduces network overhead by an average of 30% thanks to optimizing segment transmission via IPFS.

All these measurements were performed with a video server running Ubuntu, an Intel Core i7 2.8 GHz, 8 GB RAM, and a set of 20 to 60 simulated MEC nodes running Omnet++ framework. As a first step, we are building the fundamental components of the system architecture using open source frameworks such as INET and Openflow [137], as well as cryptography libraries including OpenSSL [138] and CryptoPP [139]. We are using the INET framework to implement simulation models integrating Internet protocols such as IPV4, TCP, UDP, BGP, and OSPFS, making it particularly well suited to video surveillance applications.

The security assessment is also based on a series of experimental scenarios designed to evaluate the performance of the proposed architecture. Each of these scenarios integrates an architectural simulation model based on the OpenFlow framework. These models also create an SDN control environment intended to optimize, evaluate, and test the level of security within the edge layer while taking into consideration the associated costs. In parallel, we developed a simulation model based on the INET framework, specifically designed for organizations, IPFS nodes, and the video server. Thanks to this prototype, we were able to run tests and evaluate the application performance in various load and network traffic configurations. This analysis proved to be of crucial importance in anticipating the evolution of a video surveillance application as a function of network size.

4.6.1 Smart contract implementation

The smart contract is implemented in the Go language, whose interactions with the edge layer nodes are defined by UDP sockets and a REST API via the HTTP protocol. The source code example illustrates the function of adding a VCID used by authorized IPFS nodes, managing authorizations, and triggering events whenever VCIDs are added.

```
package main
import (
    "github.com/smartcontractkit/chainlink/solidity/
    v0.4/chainlink"
    "math/big"
    ...
)

// Video Surveillance Security Smart Contract
contract SecurityVideoContract {
    // State variables
    // Contract owner
    address public owner;
    // Autorized edge nodes
    mapping(address => bool)
```

Table 4.3: Simulation settings.

Parameters	Value	Description
Video size	2MB - 100MB	This range allows testing the system with videos of different sizes.
Video segment size	10Ko	Manage and transmit video data while maintaining the granularity needed for accurate validation.
video packet size	1-2 Ko	Used in networks to balance the transmission load and reduce the risk of packet loss.
Send Interval Time	5ms	Regular data transmission, making it possible to test the system's ability to manage continuous video streams.
Hash function	SHA-256	Ensures data integrity and security.
Block validation function	SHA-256	Block validation, consistency, strengthening the security of the blockchain.
NIC bitrate	4 Mbps	This rate is chosen to simulate the realistic capabilities of network cards in edge computing environments.
Data rate	100Mbps	A high data rate to represent modern networks, testing the system performance under heavy loads.
Number of Edge Nodes	20-60	Test different levels of decentralization, load and scalability.
Number of video server	1	A single video server represents a central distribution point.
Number of camera	1	A single camera is used.
Number of mobile users	1-50	Simulating a variable number of mobile users under different load levels.
Storage latency	5ms	High-performance storage delay, reflecting the real capacities of modern infrastructures.
Transmission protocol	UDP	Ensure low latency and effective transmissions.
Transmission latency	5ms	This latency reflects realistic transmission delays in various networks.
Transmission success rate	99%	A high success rate is realistic for well-configured modern networks, while also testing the robustness of the system in the face of minor failures.
Organisation/Edge processing delay	10ms	Data processing time delay by the edge nodes.
Organisation/Edge nodes delay	1.25E-4us	This extremely short delay reflects the fast-processing capabilities of edge nodes.

```
    public authorized_Edge_Nodes;
    // Autorized organisations
    mapping(address => bool)
    public authorized_Organisations;
    // Video data counter
    uint256 public videoDataCount;

    // Event triggered when new video data is added
    event VideoDataAdded(address indexed IpfsNode
    , string VCID, uint256 timestamp);

    // Modifier to restrict access to authorized
    organisations only
    modifier onlyAuthorizedOrganisation() {
        require(authorizedOrganisations[msg.sender],
        "The organisation is not authorized");
    }

    // Contract constructor
    constructor() {
        ...
    }

    // Function to add video data
    function addVideoData(string memory VCID) public
    onlyAuthorizedIPFSNode {
        videoDataCount++;
        emit VideoDataAdded(msg.sender, VCID,
        block.timestamp);
}
```

The following queries shows two examples of smart contract events via the REST API, along with their corresponding responses. The first example concerns a request to add a VCID to the blockchain, while the second concerns a request to obtain a storage registration ID from IPFS nodes.

Adding a VCID via POST request:

```
POST /api/addVCID Content-Type:
application/json
{
"ipfsNodeAddress": "0xIpfsNodeAddress",
"VCID": "0xVCID"
}
```

REPLAY : Case of authorized IPFS node

```
{
  "message": "The VCID was added successfully.",
  "videoDataCount": 1
}
```

REPLAY : Case of unauthorized IPFS node

```
{
  "error": "The IPFS node is not authorized to perform
  this action."
}
```

Storage request registration ID via GET request:

```
GET /api/getStorageRegistrationID?
IpfsNodeAddress=0xIpfsNodeAddress
```

REPLAY : Case of authorized IPFS node

```
{
  "StorageRegistrationID": a3fd933ba9cdd
}
```

REPLAY : Case of unauthorized IPFS node

```
{
  "error": "The ipfs node is not authorized to access
  this resource."
}
```

4.6.2 Security performances

In this subsection, we analyze in detail the results obtained during our simulations, highlighting the performance of our solution in terms of storage capacity and security overhead. The objective of this analysis is to highlight the ability of different blockchains to evolve according to their size and the signature algorithms they use. Our focus is on the SegWit blockchain and its performance compared to other security models. This comparison will help us assess the ability of each model to handle blockchain growth while identifying the advantages of the proposal over other approaches in terms of scalability and efficiency.

Storage capacity

Initially, we focus on the evaluation of storage capacity as the size of the blockchain increases while preserving security. More specifically, we compare the increase shown by our model against other equivalent security models. The figure 4.5 shows the increase in SegWit blockchain size compared with a blockchain model adopting an RSA signature verification system and a key size of 4096 bits (simulating the Zcoin video surveillance protocol), as well as a blockchain

model using ECDSA signatures with a P-256 curve. In addition, the figure 4.6 shows the transaction speed histogram to demonstrate the speed of the proposed model. In which, another key indicator evaluated in our simulation is the number of transactions our system can process per second (TPS). We compared our SegWit model to RSA and ECDSA approaches. We carry out a scenario involving a surveillance video of size 10 MIB to evaluate the size of the blockchain, measured in bytes, and the number of TPS for the three models we are examining. In this context, the SegWit blockchain adopts a 128-bit signature verification key, while the ECDSA-based blockchain model adopts a 512-bit signature verification key. Similarly, the RSA-based blockchain model uses a 4096-bit signature verification key. The results indicate that the SegWit structure occupies 1.35KB of storage space, compared with the ECDSA-based blockchain model, which requires 2.1KB of storage space, and the RSA-based blockchain model, which requires 7.1KB of storage space. In terms of transaction speed, the SegWit blockchain can process up to 7 transactions per second, while the ECDSA-based blockchain achieves a speed of 4 transactions per second, and the RSA-based blockchain model can only generate one transaction per second.

The simulation results show that the SegWit implementation reduces the size of transactions stored on the blockchain compared to traditional solutions based on RSA and ECDSA. Specifically:

- The block size in our SegWit approach reaches 1.35 KB, compared to 2.1 KB for an ECDSA-P256 model and 7.1 KB for RSA-4096.
- This reduction in memory overhead directly contributes to improved system scalability, enabling more efficient storage on MEC nodes.
- SegWit blockchain transaction rate per second reaches 7 TPS faster than ECDSA-P256 model (4 TPS) and RSA-4096 model (TPS)

These results confirm the storage efficiency, memory usage, and scalability of the proposed SegWit blockchain with its integrity check scheme compared to other models. We observe that the video surveillance application using the SegWit structure meets the increasing storage demands of blockchain in a MEC environment. While the RSA-based blockchain model for transaction verification is relatively large, for a video surveillance application, this blockchain could be used if security is a top priority and size efficiency is less critical.

In addition, the improved transaction speed is due to the optimized storage of signatures and integrity proofs in SegWit blocks, enabling faster verification while reducing block size. This shows that our model is more suitable for environments requiring fast and secure validations, such as MEC video surveillance networks.

Security overhead

To assess the overhead of security on network performance during the storage and transmission processes, we carried out a comparison of three distinct scenarios in order to measure the latency

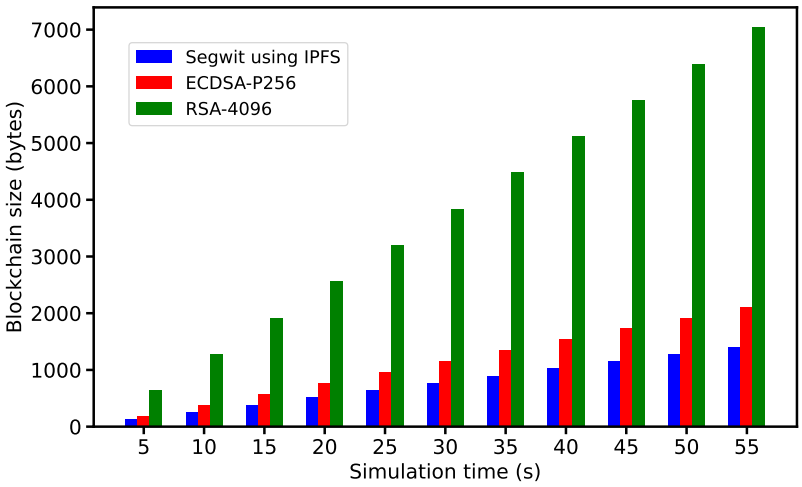


Figure 4.5: The security performances.

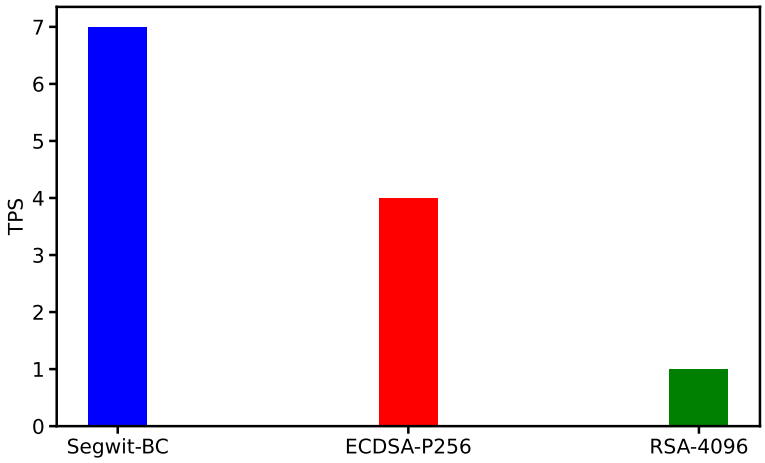


Figure 4.6: The transaction speed.

associated with the storage and transmission processes. The first scenario concerns a distributed video storage system (SWARM) based on the Hyperledger Fabric blockchain [136]. This model relies on ECC signature verification to guarantee the integrity of video data. The second scenario concerns a cloud-based intelligent surveillance system [135], where the integrity of stored videos is verified using a root hash provided by Merkle tree. The figure 4.7 shows the observed storage latency and transmission latency of the proposed approach. We can see that the data storage process precedes the transmission process. This sequence is due to the security load that transactions undergo in the blockchain as well as their verification process. It is also worth noting that the latency of storage transactions has a range of 50 to 50.03ms, while the latency of transmission transactions has an average range of 50.04 to 50.06ms.

The figure 4.8a shows an evaluation of storage performance for different video sizes. We have compared the performance of the IPFS storage process in our model with that of the

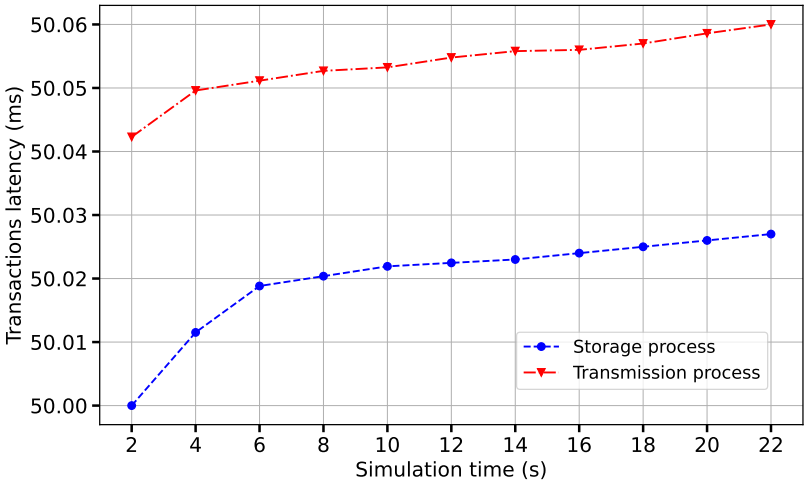


Figure 4.7: The latency of storage and transmission transactions.

distributed video storage process (SWARM) based on the Hyperledger Fabric blockchain, as well as with cloud-based intelligent storage, which verifies the integrity of stored videos using a root hash provided by Merkle tree. As a result, the IPFS storage process in our model shows a latency of 50s for a 100 MIB video, while the SWARM storage process achieves a latency of around 70s. On the other hand, cloud-based storage using Merkle tree has a much higher latency, reaching 100s. These results highlight the efficiency of the storage process proposed by our model compared with reference models. In particular, our approach features faster storage performance, which is essential in scenarios where latency is a critical factor for video data availability and access. Similarly, the figure 4.8b shows the transmission performance, including the verification process, for different video sizes within the IPFS system (edge layer) at the organization. We compare the performance of the transmission process of our proposed system with the distributed video transmission process (SWARM) based on the Hyperledger Fabric blockchain, including the verification process. Initially, we find that SWARM transmission outperforms our proposal in a preliminary stage. However, the results underline that our system exhibits increasingly efficient latency as the size of the video increases.

Consequently, we measured the video storage and transmission latency in our system, comparing our approach with SWARM (Hyperledger Fabric) and a Merkle Tree-based cloud storage solution. Simulation results show that:

- Storage latency in our IPFS + SegWit model is 50 ms for a 100 MB video, compared to 70 ms for Hyperledger Fabric and 100 ms for Merkle Tree cloud storage.
- Transmission latency follows a similar trend: our solution reduces video transmission time by 20 to 30%, notably thanks to optimized routing via IPFS and avoiding network congestion through video segmentation.

These results confirm that our solution is more efficient in terms of transmission and

bandwidth management, which is essential for real-time applications such as distributed video surveillance.

The figure 4.8c shows the time required to verify the integrity of the proposed blockchain in comparison with the Hyperledger Fabric blockchain model. The results show that, for small videos, the verification time of the Hyperledger Fabric blockchain is lower than that of our model. However, as video size increases, the verification time of our blockchain model decreases compared to the Hyperledger-based model. The figure 4.8d illustrates the video segment integrity verification time, ranging from 5 to 50 segments, verified by our model, as well as the verification time by hash tree, and finally, the verification time according to an RSA verification scheme. The RSA scheme shows a verification time of 3.03s to 12.33s, while hash tree show a verification time ranging from 0.72s to 3.34s. On the other hand, our model provides a faster and more efficient verification process, with a time ranging between 0.64s and 2.55s, compared to other models.

Overall, the time required to validate 50 video segments compared to a Merkle tree-based model and an RSA approach:

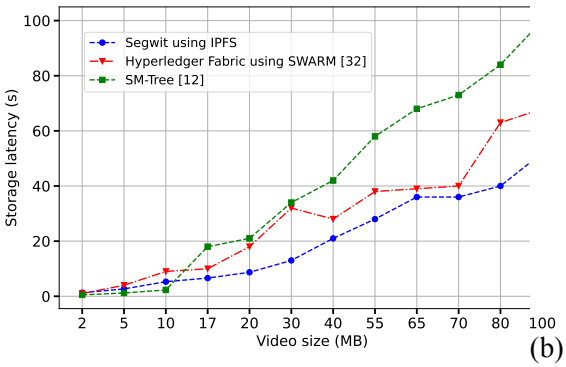
- Our solution (SegWit + IPFS): 0.64 s to 2.55 s
- Merkle tree: 0.72 s to 3.34 s
- RSA: 3.03 s to 12.33 s

We find that our approach reduces verification time by 20% compared to a Merkle tree and is four times faster than RSA, demonstrating the effectiveness of our SegWit-based validation mechanism.

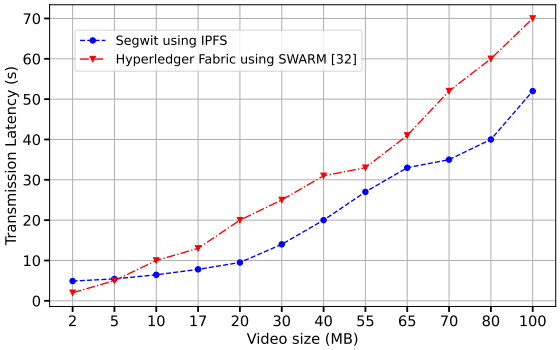
The figures 4.8e and 4.8f show the scalability of the proposed security model, comparing performance with and without SDN architecture. The plots highlight the diversity of latencies as a function of the number of organizations. In the presence of an SDN architecture, we observe that the latency of our security model varies in milliseconds, remaining stable for the first 20 organizations (52.45ms), then increasing significantly until it reaches 60.40ms for 100 organizations. By contrast, in the absence of SDN architecture, the latency of our security model is expressed in seconds, remaining stable for the top 50 organizations (2.5s), then increasing significantly to 4.23s for 100 organizations. These results illustrate the network performance efficiency of our approach compared with traditional VSS generations.

4.7 Discussion

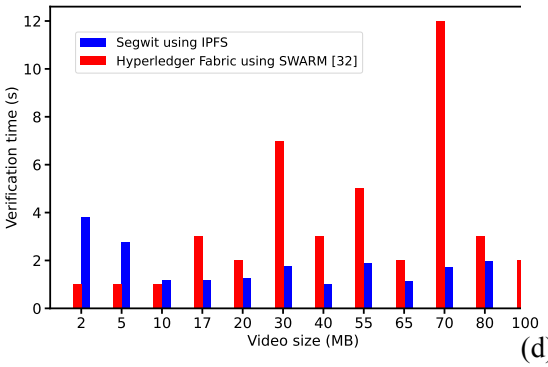
In this section, we provide a comprehensive discussion and analysis of the network performance and overall experimental results of our proposed approach. We compare our approach against several state-of-the-art methods to highlight its superiority in terms of integrity, privacy



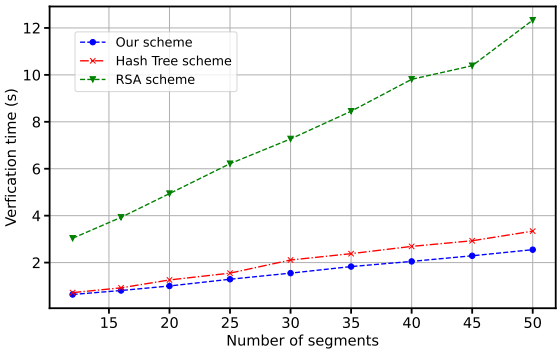
(a) The latency of the video storage process.



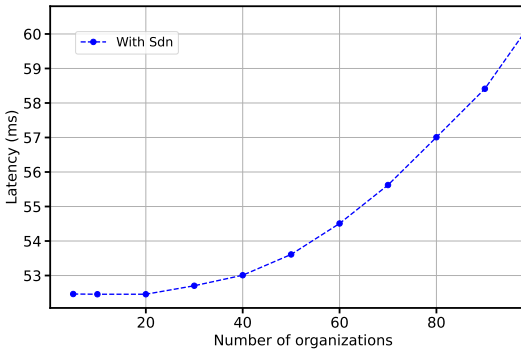
(b) The latency of the video transmission process.



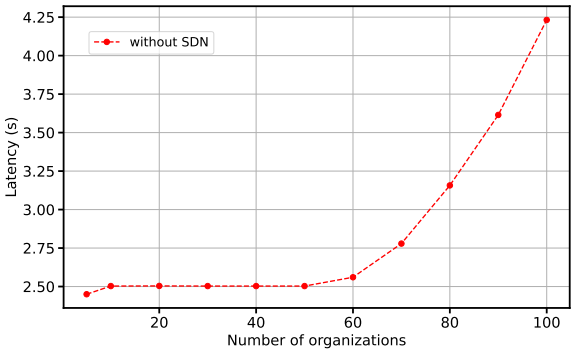
(c) The integrity verification time.



(d) The verification time of the proposed schema.



(e) The scalability over SDN networks.



(f) The scalability without SDN networks.

Figure 4.8: The network performances.

protection, level of distribution, security level, lightweight performance, memory usage, security overhead, and scalability. The table 4.4 provides a comparative analysis of various VSS approaches based on these criteria.

Criterion	Our Approach	Merkle Tree-Based Method [135]	EAMSuS [141]	Video Forensics Framework [140]
Integrity	Utilizes blockchain and SegWit for robust video data integrity verification.	Ensures data integrity via Merkle trees, but with limited capabilities.	Focus on media processing efficiency, less emphasis on integrity.	Strong verification of video evidence authenticity via blockchain.
Privacy Protection	Integrates advanced encryption techniques and smart contracts for privacy.	Protects privacy using masking and encryption techniques.	Less emphasis on privacy protection, focuses on processing efficiency.	Strong protection of the privacy of video evidence.
Level of Distribution	Highly distributed through IPFS and MEC, enhancing data availability and redundancy.	Less distributed, primarily depends on data deduplication.	Limited distribution within IoT environments, less focus on overall distribution.	Effective distribution for legal applications, less focus on generalized distribution.
Security Level	High level of security with blockchain, SegWit, and smart contracts.	Security ensured by Merkle trees, but less robust than blockchain approach.	Basic security with a focus on processing efficiency.	High security with blockchain for evidence authenticity.
Lightweight Performance	Optimized performance with MEC, reducing processing load and minimizing latency.	Moderate performance, potentially limited without edge computing support.	Efficient performance in processing, not optimized for storage and transmission.	Less focus on lightweight performance, more on verification and authenticity.
Memory Usage	Efficient memory usage with SegWit, reducing transaction sizes.	Reduced memory usage through data deduplication, but without edge computing, it can be limited.	Moderate memory usage with a focus on processing efficiency.	Less optimized memory usage, more oriented towards proof verification.
Security Overhead	Reduced overhead through SegWit optimization and distribution via MEC and IPFS.	Moderate overhead, potentially high without edge computing support.	Low overhead but with limited security mechanisms.	High overhead for integrity and authenticity verification.
Scalability	Highly scalable with IPFS and MEC, efficiently managing increasing video data volumes.	Limited scalability without edge computing support, can become inefficient at large scale.	Scalable within IoT environments, may face challenges in broader deployments.	Scalable for legal applications, but less optimized for general video surveillance data handling.

Table 4.4: Comparative Analysis of Different Approaches

In terms of network performance, our proposed approach integrates MEC layer and SegWit blockchain to optimize the storage and transmission of video data. This integration significantly reduces processing loads on nodes, minimizes latency, and enhances overall network performance. Experimental results demonstrate that our system can handle large volumes of video files more efficiently than the Merkle Tree-Based Method, EAMSuS, and the Video Forensics Framework. Specifically, the use of SegWit in our blockchain structure improves transaction throughput and scalability, making it more suitable for large-scale deployments.

From integrity and privacy protection perspective, the proposed system ensures robust video data integrity verification through the combination of blockchain and SegWit. This approach outperforms the Merkle Tree-Based Method, which, while effective, offers limited capabilities without the support of edge computing. Additionally, our method integrates advanced encryption techniques and smart contracts, providing superior privacy protection compared to the other methods which focus less on privacy.

Regarding the distribution and security level, the high level of distribution achieved through

IPFS and MEC in our approach enhances data availability and redundancy. This is a significant improvement over the Merkle Tree-Based Method, which primarily relies on data duplication and lacks edge computing support. Furthermore, our approach provides a high level of security by combining blockchain, SegWit, and smart contracts, making it more robust than methods using only Merkle trees or basic security mechanisms.

For lightweight performance and memory usage, our system is optimized for lightweight performance with MEC, reducing processing loads and minimizing latency. This makes it more efficient than methods not specifically optimized for storage and transmission, such as EAMSuS. Additionally, the efficient memory usage achieved through SegWit, which reduces transaction sizes, is a significant advantage over other methods that may become limited without edge computing support.

In terms of security and scalability overhead, our approach reduces security overhead by leveraging SegWit optimization and distribution via MEC and IPFS. Which makes our approach more efficient than other models that can incur higher fees without edge computing support. Our system is also highly scalable, efficiently handling increasing volumes of video data. This scalability is superior to Merkle Tree-Based and other approaches that may struggle with larger deployments without additional support.

As a result, our approach excels in several areas by utilizing blockchain and SegWit for robust video data integrity verification, integrating advanced encryption techniques and smart contracts for enhanced privacy protection, and ensuring a high level of data distribution through IPFS and MEC, which enhances availability and redundancy. In terms of security, our approach offers a superior level of protection with blockchain, SegWit, and smart contracts, compared to the more basic security measures of other methods. Performance is optimized with MEC, reducing processing load and minimizing latency, making it more efficient than methods not optimized for storage and transmission. Memory usage is efficiently managed by reducing transaction sizes with SegWit. Additionally, security overhead is minimized through SegWit optimization and MEC/IPFS distribution. Our approach is also highly scalable, efficiently managing increasing video data volumes. This comprehensive solution outperforms other state-of-the-art methods in terms of network performance, security, and scalability.

This comparative analysis clearly demonstrate the superiority of our proposed approach in terms of network performance, security, and scalability. By integrating MEC, blockchain, and SegWit, our system offers a comprehensive solution that outperforms existing methods, ensuring robust data integrity, enhanced privacy protection, efficient memory usage, reduced security overhead, and high scalability. This makes our approach the optimal choice for modern VSS.

4.8 Perspectives Futures

The results obtained in this study pave the way for several future research directions to improve and extend VSS in MEC environments:

4.8.1 Identified Limitations and Future Improvements

While our proposed approach enhances video data security and network performance, certain limitations need to be addressed in future research:

- **Computational and Storage Overhead:** The integration of blockchain in MEC environments introduces additional computational costs and increased storage demands on edge nodes. Future work could focus on optimizing resource allocation strategies and exploring lightweight blockchain architectures to reduce this overhead.
- **Scalability in Congested Networks:** Although our system improves data transmission efficiency, its performance may degrade in highly congested network environments. Investigating adaptive data routing mechanisms and dynamic load balancing strategies would help mitigate this issue.
- **Dependence on MEC and IPFS Availability:** The effectiveness of our system relies on the continuous availability of MEC nodes and IPFS storage. Future research could explore decentralized redundancy techniques and fault-tolerant architectures to enhance reliability in unstable network conditions.
- **Security Threats:** While our approach strengthens security, it remains susceptible to Sybil attacks and Distributed Denial of Service (DDoS) attacks on MEC nodes. Future enhancements could include the integration of AI-based anomaly detection and blockchain-based identity verification mechanisms.

4.8.2 Improved Scalability and Security

A future research direction could be to explore the use of new consensus protocols that provide better scalability and enhanced security. For example, combining Proof of Stake (PoS) with sharding mechanisms could further improve system performance while reducing the computational burden. Additionally, the integration of post-quantum encryption technologies can be explored to protect video data against future threats posed by quantum computers.

4.8.3 Performance Optimization of Peripheral Nodes

Another promising avenue of research is the integration of AI capabilities directly into edge nodes to perform real-time analysis of video streams, which would reduce latency and load on the core network. Furthermore, developing energy management strategies to optimize the energy consumption of edge nodes would ensure sustainable and efficient operation, addressing concerns about computational overhead.

4.8.4 Extension of Coverage and Interoperability

To improve interoperability, it would be beneficial to work on standards and protocols allowing seamless integration between different VSS in existing infrastructures. Additionally, leveraging the capabilities of 5G networks and future networks beyond 5G could improve bandwidth, reduce latency, and increase the number of connected devices supported by the system, thereby mitigating scalability issues in congested environments.

4.8.5 Case Studies and Real Deployments

It would also be valuable to implement and evaluate the systems in real-world environments such as critical infrastructure, university campuses, or smart cities to validate the effectiveness and efficiency of the proposed solutions. Collecting and analyzing feedback from end users will help identify areas for improvement and refine technological solutions, particularly in terms of system reliability and resistance to cyber threats.

Chapitre 4

Conclusion

In this chapter, we proposed a MEC architecture that integrates blockchain to secure the storage and transmission of video surveillance data. This architecture is based on four essential components: the video server, IPFS nodes, the SegWit blockchain, and participating organizations. MEC nodes enable the implementation of a decentralized video storage system via IPFS, where each video segment is identified by a Video Content Identifier (VCID) and accompanied by a proof of integrity validated by the SegWit blockchain. To evaluate the effectiveness of our approach, we conducted a series of experimental tests on a simulation prototype, comparing it to existing solutions. The results show that our solution:

- Optimizes storage through to the integration of SegWit, reducing the size of transactions on the blockchain.
- Improves the transactions per second (TPS) rate compared to models using ECDSA and RSA.
- Reduces storage and transmission latency, particularly compared to Hyperledger Fabric and Merkle Tree-based cloud solutions.
- Guarantees fast and secure validation of the integrity of videos stored and transmitted via IPFS.

While these results are promising, our approach has some limitations, especially in terms of network load in highly constrained environments. Future optimizations could be considered to further reduce resource consumption and improve system scalability. This performance indicates that the integration of SegWit, IPFS, and MEC provides an efficient and scalable solution, suitable for modern video surveillance infrastructures, while balancing security, scalability, and network performance.

General Conclusion

The rise of the Mobile Edge Computing (MEC) paradigm in modern network and data processing architectures offers a direct response to the growing demands for low latency, bandwidth optimization, data security, and service personalization. This thesis aligns with this technological and scientific evolution by proposing innovative solutions to secure and optimize data processing, transfer, and storage within distributed, mobile, and heterogeneous environments.

Our work began with an in-depth analysis of quality of service (QoS) and security challenges in MEC environments. Through a systematic review of enabling technologies — notably Software Defined Networking (SDN), Network Functions Virtualization (NFV), Blockchain, and the InterPlanetary File System (IPFS) — we demonstrated the synergy among these components in building robust, flexible, and intelligent MEC-based architectures. This study also highlighted the limitations of conventional models, especially regarding centralized resource management, handling critical traffic flows, and protecting sensitive data.

From this theoretical foundation, the thesis proposed three key contributions, each addressing real-world use cases:

1. **Securing Mobile Data Offloading in Small Cell Networks** : Our first contribution introduced a secure architecture for mobile data offloading via small cell networks. This solution leveraged a combination of distributed SDN controllers and both public and private blockchain systems, enabling smart and secure selection of transmission paths (Wi-Fi, D2D, Small Cells) based on trust level, network load, and data sensitivity. Our approach successfully balanced low latency, high network robustness, and confidentiality guarantees, as confirmed through a series of QoS and security-focused simulations.

2. **Intelligent Optimization of Security Task Management**: The second contribution tackled the challenge of managing security-related tasks in resource-constrained MEC environments. We proposed a distributed task orchestration model based on NFV, where security functions are dynamically assigned to edge nodes via a NFV Orchestrator (NFVO). Compared to heuristic and priority-based approaches, our solution showed superior response times, load balancing, and computational efficiency. This is particularly relevant for real-time streaming, connected vehicles, and mission-critical IoT applications.

3. **Secure and Decentralized Video Surveillance Storage with MEC + IPFS + Blockchain**: Our third contribution focused on the fast-growing domain of connected video surveillance.

We developed a hybrid architecture combining IPFS for distributed storage, blockchain (with SegWit) for integrity verification, and smart contracts for autonomous governance. This architecture ensures resilience to network outages, traceability of data, and legal-grade authenticity of stored video evidence. Experimental results validated the effectiveness of this solution in terms of latency, integrity, scalability, and security compared to conventional cloud-based systems.

Limitations and Lessons Learned : Although the results are promising, some limitations must be acknowledged. Firstly, the computational and energy costs associated with technologies such as blockchain still present a challenge for large-scale deployment. More lightweight consensus mechanisms or consortium-based blockchain implementations could help mitigate this.

Secondly, the lack of interoperable standards for harmonizing SDN, NFV, and blockchain in MEC environments is still a barrier to widespread adoption. Moreover, real-world implementation challenges such as physical edge node security, software updates, and fault tolerance require further attention.

Privacy regulation (e.g., GDPR compliance), legal validation of video evidence, and trust management in autonomous edge devices are also important aspects that call for deeper exploration. Lastly, while our simulations and experiments were comprehensive, they should eventually be complemented by real-world pilot deployments for further validation.

Future Research Directions: This thesis opens up several promising research perspectives:

- **Toward Edge Intelligence (EI):** Integrating machine learning at the edge can enable dynamic adaptation of security policies, anomaly detection, and real-time network optimization.
- **Smart Contract Automation:** Beyond access control, smart contracts could be used for dynamic pricing, trust negotiation, and autonomous inter-device coordination in IoT networks.
- **MEC in the 6G Era:** As we move toward 6G, MEC architectures must evolve to support ultra-low latency, intelligent distributed services, and new paradigms such as holographic networks or brain–computer interfaces.
- **Testbeds and Real-World Trials:** Further evaluation should be carried out in smart city pilots, critical infrastructure, or intelligent transportation systems to demonstrate the operational effectiveness of our solutions under real constraints.

- International journals with peer review committee

1) Cheriet, A., & Mekhaznia, T. (2025). Optimizing network performance and security in video surveillance data storage: a solution with blockchain and mobile edge computing. Computing, 107(6), 1-28.

- International conferences with reading committee

1) Cheriet, A., & Mekhaznia, T. (2024, April). Intelligent optimization of computing task management in an edge environment. In 2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS) (pp. 1-7). IEEE..
2) Cheriet, A., & Mekhaznia, T. (2022, October). Secure Mobile Data Offloading in Small Cell Networks. In 2022 4th International Conference on Pattern Analysis and Intelligent Systems (PAIS) (pp. 1-6). IEEE.

Bibliography

[1] Sehgal, N. K., Bhatt, P. C. P., & Acken, J. M. (2020). Cloud computing with security and scalability. Springer, https://link. springer. com/book/10.1007/978-3-031-07242-0.

[2] Kulkarni, G., Sutar, R., & Gambhir, J. (2012). Cloud computing-Infrastructure as service-Amazon EC2. International journal of Engineering research and applications, 2(1), 117-125.

[3] Li, G., Wang, J., Wu, J., & Song, J. (2018). Data processing delay optimization in mobile edge computing. Wireless Communications and Mobile Computing, 2018(1), 6897523.

[4] Chen, X., Cai, Y., Li, L., Zhao, M., Champagne, B., & Hanzo, L. (2019). Energy-efficient resource allocation for latency-sensitive mobile edge computing. IEEE Transactions on Vehicular Technology, 69(2), 2246-2262.

[5] Sun, X., & Ansari, N. (2016). EdgeIoT: Mobile edge computing for the Internet of Things. IEEE Communications Magazine, 54(12), 22-29.

[6] Qin, L., Lu, H., & Wu, F. (2022). When the user-centric network meets mobile edge computing: Challenges and optimization. IEEE Communications Magazine, 61(1), 114-120.

[7] Costa-Requena, J., Santos, J. L., Guasch, V. F., Ahokas, K., Premsankar, G., Luukkainen, S., ... & de Oca, E. M. (2015, June). SDN and NFV integration in generalized mobile network architecture. In 2015 European conference on networks and communications (EuCNC) (pp. 154-158). IEEE.

[8] Han, K., Li, S., Tang, S., Huang, H., Zhao, S., Fu, G., & Zhu, Z. (2018). Application-driven end-to-end slicing: When wireless network virtualization orchestrates with NFV-based mobile edge computing. IEEE Access, 6, 26567-26577.

[9] Feng, J., Pei, Q., Yu, F. R., Chu, X., Du, J., & Zhu, L. (2020). Dynamic network slicing and resource allocation in mobile edge computing systems. IEEE Transactions on Vehicular Technology, 69(7), 7863-7878.

[10] Zhang, X., Li, Z., Lai, C., & Zhang, J. (2021). Joint edge server placement and service placement in mobile-edge computing. IEEE Internet of Things Journal, 9(13), 11261-11274.

[11] Naouri, A., Wu, H., Nouri, N. A., Dhelim, S., & Ning, H. (2021). A novel framework for mobile-edge computing by optimizing task offloading. IEEE Internet of Things Journal, 8(16), 13065-13076.

[12] Liang, Z., Liu, Y., Lok, T. M., & Huang, K. (2021). Multi-cell mobile edge computing: Joint service migration and resource allocation. IEEE Transactions on Wireless Communications, 20(9), 5898-5912.

[13] Zhang, K., Leng, S., He, Y., Maharjan, S., & Zhang, Y. (2018). Cooperative content caching in 5G networks with mobile edge computing. IEEE Wireless Communications, 25(3), 80-87.

[14] Sun, Y., Zhou, S., & Xu, J. (2017). EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks. IEEE Journal on Selected Areas in Communications, 35(11), 2637-2646.

[15] Zhang, K., Mao, Y., Leng, S., Zhao, Q., Li, L., Peng, X., ... & Zhang, Y. (2016). Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks. IEEE access, 4, 5896-5907.

[16] Ranaweera, P., Jurcut, A. D., & Liyanage, M. (2021). Survey on multi-access edge computing security and privacy. IEEE Communications Surveys & Tutorials, 23(2), 1078-1124.

[17] Lee, K., Lee, J., Yi, Y., Rhee, I., & Chong, S. (2012). Mobile data offloading: How much can WiFi deliver?. IEEE/ACM Transactions on networking, 21(2), 536-550.

[18] Dai, Y., Xu, D., Maharjan, S., & Zhang, Y. (2018). Joint computation offloading and user association in multi-task mobile edge computing. IEEE Transactions on Vehicular Technology, 67(12), 12313-12325.

[19] Chen, L., Zhou, S., & Xu, J. (2018). Computation peer offloading for energy-constrained mobile edge computing in small-cell networks. IEEE/ACM transactions on networking, 26(4), 1619-1632.

[20] Cheng, R. G., Chen, N. S., Chou, Y. F., & Becvar, Z. (2015). Offloading multiple mobile data contents through opportunistic device-to-device communications. Wireless Personal Communications, 84, 1963-1979.

[21] Elgendy, I. A., Zhang, W., Tian, Y. C., & Li, K. (2019). Resource allocation and computation offloading with data security for mobile edge computing. Future Generation Computer Systems, 100, 531-541.

[22] Lee, J. H., Singh, K. D., Bonnin, J. M., & Pack, S. (2013). Mobile data offloading: a host-based distributed mobility management approach. IEEE Internet Computing, 18(1), 20-29.

[23] Dai, B., Niu, J., Ren, T., & Atiquzzaman, M. (2022). Toward mobility-aware computation offloading and resource allocation in end–edge–cloud orchestrated computing. IEEE Internet of Things Journal, 9(19), 19450-19462.

[24] Amani, M., Mahmoodi, T., Tatipamula, M., & Aghvami, H. (2014). SDN-based data offloading for 5G mobile networks. ZTE communications, 12(2), 34-40.

[25] Hassija, V., Saxena, V., & Chamola, V. (2021). A mobile data offloading framework based on a combination of blockchain and virtual voting. Software: Practice and Experience, 51(12), 2428-2445.

[26] Alrowais, F., Almasoud, A. S., Marzouk, R., Al-Wesabi, F. N., Hilal, A. M., Rizwanullah, M., ... & Yaseen, I. (2022). Artificial Intelligence Based Data Offloading Technique for Secure MEC Systems. Computers, Materials & Continua, 72(2).

[27] Sanvito, D., Moro, D., & Capone, A. (2017, July). Towards traffic classification offloading to stateful SDN data planes. In 2017 IEEE Conference on Network Softwarization (NetSoft) (pp. 1-4). IEEE.

[28] Jiang, C., Cheng, X., Gao, H., Zhou, X., & Wan, J. (2019). Toward computation offloading in edge computing: A survey. IEEE Access, 7, 131543-131558.

[29] Tong, Z., Deng, X., Ye, F., Basodi, S., Xiao, X., & Pan, Y. (2020). Adaptive computation offloading and resource allocation strategy in a mobile edge computing environment. Information Sciences, 537, 116-131.

[30] Open Networking Foundation (ONF). [Online]. Available: https://www.opennetworking.org/

[31] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., ... & Turner, J. (2008). OpenFlow: enabling innovation in campus networks. ACM SIGCOMM computer communication review, 38(2), 69-74.

[32] ONF (Open Networking Foundation). (2012). Software-Defined Networking: The New Norm for Networks. ONF White Paper. [En ligne] : https://opennetworking.org

[33] Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2014). Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1), 14-76.

[34] ETSI (European Telecommunications Standards Institute). (2014). Network Functions Virtualisation (NFV); Architectural Framework. ETSI GS NFV 002 V1.2.1.

[35] "Software-defined networking: The new norm for networks," Palo Alto, CA, USA, White Paper, Apr. 2012. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/white-papers/wp-sdnnewnorm.pdf

[36] ETSI, Network Functions Virtualisation (NFV); Architectural Framework, ETSI GS NFV 002 V1.2.1, European Telecommunications Standards Institute (ETSI), Sophia Antipolis, France, Dec. 2014.

[37] [2] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network Function Virtualization: State-of-the-Art and Research Challenges," IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 236–262, First Quarter 2016.

[38] [3] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network Function Virtualization: Challenges and Opportunities for Innovations," IEEE Communications Magazine, vol. 53, no. 2, pp. 90–97, Feb. 2015.

[39] ISO, Blockchain and Distributed Ledger Technologies (DLT) – Overview of and interactions between smart contracts in blockchain and DLT systems, ISO/TC 307, 2021. [Online]. Available: https://www.iso.org/committee/6266604.html

[40] ITU-T, Focus Group on Application of Distributed Ledger Technology (FG-DLT), International Telecommunication Union, 2017. [Online]. Available: https://www.itu.int/en/ITU-T/focusgroups/dlt

[41] ETSI, Permissioned Distributed Ledger (PDL); Reference Architecture, ETSI GS PDL 004 V1.1.1, 2020. [Online]. Available: https://www.etsi.org/technologies/permitted-distributed-ledger

[42] Linux Foundation, Hyperledger Project, 2022. [Online]. Available: https://www.hyperledger.org

[43] Enterprise Ethereum Alliance (EEA), Enterprise Ethereum Client Specification V7, 2021. [Online]. Available: https://entethalliance.org

[44] European Commission, European Blockchain Services Infrastructure (EBSI), 2022. [Online]. Available: https://ec.europa.eu/digital-strategy/our-policies/european-blockchain-services-infrastructure-ebsi

[45] Garg, S., Kaur, K., Kaddoum, G., Garigipati, P., & Aujla, G. S. (2021). Security in IoT-driven mobile edge computing: New paradigms, challenges, and opportunities. IEEE network, 35(5), 298-305.

[46] Mäkitalo, N., Ometov, A., Kannisto, J., Andreev, S., Koucheryavy, Y., & Mikkonen, T. (2017). Safe, secure executions at the network edge: coordinating cloud, edge, and fog computing. IEEE Software, 35(1), 30-37.

[47] Chen, Y., Qiu, Y., Tang, Z., Long, S., Zhao, L., & Tang, Z. (2024). Exploring the synergy of blockchain, IoT, and edge computing in smart traffic management across urban landscapes. Journal of Grid Computing, 22(2), 45.

[48] Elgendy, I. A., Zhang, W. Z., Zeng, Y., He, H., Tian, Y. C., & Yang, Y. (2020). Efficient and secure multi-user multi-task computation offloading for mobile-edge computing in mobile IoT networks. IEEE Transactions on Network and Service Management, 17(4), 2410-2422.

[49] Ali, B., Gregory, M. A., & Li, S. (2021). Multi-access edge computing architecture, data security and privacy: A review. Ieee Access, 9, 18706-18721.

[50] ETSI GS MEC 003 V1.1.1, "Mobile Edge Computing (MEC); Framework and Reference Architecture" (2016-03).

[51] Jararweh, Y., Doulat, A., AlQudah, O., Ahmed, E., Al-Ayyoub, M., & Benkhelifa, E. (2016, May). The future of mobile cloud computing: integrating cloudlets and mobile edge computing. In 2016 23rd International conference on telecommunications (ICT) (pp. 1-5). IEEE.

[52] Mach, P., & Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. IEEE Communications Surveys & Tutorials, 19(3), 1628-1656.

[53] Yang, M., Li, Y., Jin, D., Su, L., Ma, S., & Zeng, L. (2013). OpenRAN: a software-defined ran architecture via virtualization. ACM SIGCOMM computer communication review, 43(4), 549-550.

[54] Phan, L. A., Nguyen, D. T., Lee, M., Park, D. H., & Kim, T. (2021). Dynamic fog-to-fog offloading in SDN-based fog computing systems. Future Generation Computer Systems, 117, 486-497.

[55] Elgendy, I. A., Muthanna, A., Hammoudeh, M., Shaiba, H., Unal, D., & Khayyat, M. (2021). Advanced deep learning for resource allocation and security aware data offloading in industrial mobile edge computing. Big Data, 9(4), 265-278.

[56] Hao, Y., Cao, J., Wang, Q., & Du, J. (2021). Energy-aware scheduling in edge computing with a clustering method. Future Generation Computer Systems, 117, 259-272.

[57] Rahman, A., Islam, M. J., Montieri, A., Nasir, M. K., Reza, M. M., Band, S. S., ... & Mosavi, A. (2021). Smartblock-sdn: an optimized blockchain-sdn framework for resource management in IoT. IEEE Access, 9, 28361-28376.

[58] S. R. Basnet and S. Shakya, "BSS: Blockchain security over Software-Defined Network", International Conference on Computing Communication and Automation (ICCCA), pp. 720-725, 2017.

[59] Zhang, D., Gong, C., Zhang, T., Zhang, J., & Piao, M. (2021). A new algorithm of clustering AODV based on edge computing strategy in IOV. Wireless Networks, 27(4), 2891-2908.

[60] Bute, M. S., Fan, P., Liu, G., Abbas, F., & Ding, Z. (2021, April). A Collaborative Task Offloading Scheme in Vehicular Edge Computing. In 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring) (pp. 1-5). IEEE.

[61] Slamnik-Kriještorac, N., de Resende, H. C. C., Donato, C., Latré, S., Riggio, R., & Marquez-Barja, J. (2020, January). Leveraging mobile edge computing to improve vehicular communications. In 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC) (pp. 1-4). IEEE.

[62] Ayaz, F., Sheng, Z., Tian, D., & Guan, Y. L. (2020). A Proof-of-Quality-Factor (PoQF)-Based Blockchain and Edge Computing for Vehicular Message Dissemination. IEEE Internet of Things Journal, 8(4), 2468-2482.

[63] Nkenyereye, L., Adhi Tama, B., Shahzad, M. K., & Choi, Y. H. (2019). Secure and blockchain-based emergency driven message protocol for 5G enabled vehicular edge computing. Sensors, 20(1), 154.

[64] Zhang, J., Zhong, H., Cui, J., Tian, M., Xu, Y., & Liu, L. (2020). Edge computing-based privacy-preserving authentication framework and protocol for 5G-enabled vehicular networks. IEEE Transactions on Vehicular Technology, 69(7), 7940-7954.

[65] Qiu, X., Liu, L., Chen, W., Hong, Z., & Zheng, Z. (2019). Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing. IEEE Transactions on Vehicular Technology, 68(8), 8050-8062.

[66] Mansouri, Y., & Babar, M. A. (2021). A review of edge computing: Features and resource virtualization. Journal of Parallel and Distributed Computing, 150, 155-183.

[67] Liu, Y., Yan, J., & Zhao, X. (2022). Deep Reinforcement Learning based Latency Minimization for Mobile Edge Computing with Virtualization in Maritime UAV Communication Network. IEEE Transactions on Vehicular Technology.

[68] Dur, H. M. (2021). A container-based code offloading framework for mobile edge computing applications (Master's thesis, Middle East Technical University).

[69] Huang, X., Gong, S., Yang, J., Zhang, W., Yang, L., & Yeo, C. K. (2022). Hybrid market-based resources allocation in Mobile Edge Computing systems under stochastic information. Future Generation Computer Systems, 127, 80-91.

[70] Zhang, L., & Ansari, N. (2021). Optimizing the operation cost for UAV-aided mobile edge computing. IEEE Transactions on Vehicular Technology, 70(6), 6085-6093.

[71] Zhang, L., Peng, M., Wang, W., Jin, Z., Su, Y., & Chen, H. (2021). Secure and efficient data storage and sharing scheme for blockchain□based mobile□edge computing. Transactions on Emerging Telecommunications Technologies, 32(10), e4315.

[72] Adnan, M. H., Ahmad Zukarnain, Z., & Harun, N. Z. (2022). Quantum Key Distribution for 5G Networks: A Review, State of Art and Future Directions. Future Internet, 14(3), 73.

[73] Ghaznavi, M., Jalalpour, E., Salahuddin, M. A., Boutaba, R., Migault, D., & Preda, S. (2021). Content Delivery Network Security: A Survey. IEEE Communications Surveys & Tutorials, 23(4), 2166-2190.

[74] Yu, B., Zhang, X., You, I., & Khan, U. S. (2021). Efficient Computation Offloading in Edge Computing Enabled Smart Home. IEEE Access, 9, 48631-48639.

[75] Hui, H., Zhou, C., An, X., & Lin, F. (2019). A new resource allocation mechanism for security of mobile edge computing system. IEEE Access, 7, 116886-116899.

[76] Guo, F., Yu, F. R., Zhang, H., Ji, H., Liu, M., & Leung, V. C. (2019). Adaptive resource allocation in future wireless networks with blockchain and mobile edge computing. IEEE transactions on wireless communications, 19(3), 1689-1703.

[77] Elgendy, I. A., Zhang, W., Tian, Y. C., & Li, K. (2019). Resource allocation and computation offloading with data security for mobile edge computing. Future Generation Computer Systems, 100, 531-541.

[78] Khan, M. A., Baccour, E., Chkirbene, Z., Erbad, A., Hamila, R., Hamdi, M., & Gabbouj, M. (2022). A survey on mobile edge computing for video streaming: Opportunities and challenges. IEEE Access, 10, 120514-120550.

[79] Deebak, B. D., Al-Turjman, F., & Mostarda, L. (2020). Seamless secure anonymous authentication for cloud-based mobile edge computing. Computers & Electrical Engineering, 87, 106782.

[80] Huang, B., Li, Z., Tang, P., Wang, S., Zhao, J., Hu, H., ... & Chang, V. (2019). Security modeling and efficient computation offloading for service workflow in mobile edge computing. Future Generation Computer Systems, 97, 755-774.

[81] Jiang, X., Yu, F. R., Song, T.,& Leung, V. C. (2021). A survey on multi-access edge computing applied to video streaming: Some research issues and challenges. IEEE Communications Surveys & Tutorials, 23(2), 871-903.

[82] Zhang, W. Z., Elgendy, I. A., Hammad, M., Iliyasu, A. M., Du, X., Guizani, M., & Abd El-Latif, A. A. (2020). Secure and optimized load balancing for multitier IoT and edge-cloud computing systems. IEEE Internet of Things Journal, 8(10), 8119-8132.

[83] Liu, M., Yu, F. R., Teng, Y., Leung, V. C., & Song, M. (2018). Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing. IEEE Transactions on Wireless Communications, 18(1), 695-708.

[84] Watanabe, H., Hayashi, K., Sato, T., Kondo, T., & Teraoka, F. (2021). Management and network orchestration for edge/fog-based distributed data processing. Journal of information processing, 29, 640-648.

[85] Yuan, Q., Li, J., Zhou, H., Luo, G., Lin, T., Yang, F., & Shen, X. S. (2020). Cross-domain resource orchestration for the edge-computing-enabled smart road. IEEE Network, 34(5), 60-67.

[86] Zhang, J., Zhong, H., Cui, J., Tian, M., Xu, Y., & Liu, L. (2020). Edge computing-based privacy-preserving authentication framework and protocol for 5G-enabled vehicular networks. IEEE Transactions on Vehicular Technology, 69(7), 7940-7954.

[87] Wu, T. Y., Meng, Q., Yang, L., Guo, X., & Kumari, S. (2022). A provably secure lightweight authentication protocol in mobile edge computing environments. The Journal of Supercomputing, 78(12), 13893-13914.

[88] Sarier, N. D. (2021). Multimodal biometric authentication for mobile edge computing. Information Sciences, 573, 82-99.

[89] Yan, X., Wu, Q., & Sun, Y. (2020). A homomorphic encryption and privacy protection method based on blockchain and edge computing. Wireless Communications and Mobile Computing, 2020, 1-9.

[90] Li, B., He, Q., Chen, F., Jin, H., Xiang, Y., & Yang, Y. (2020). Auditing cache data integrity in the edge computing environment. IEEE Transactions on Parallel and Distributed Systems, 32(5), 1210-1223.

[91] Kim, H., Kang, E., Broman, D., & Lee, E. A. (2020). Resilient authentication and authorization for the Internet of Things (IoT) using edge computing. ACM Transactions on Internet of Things, 1(1), 1-27.

[92] Chaudhry, S. A., Alhakami, H., Baz, A., & Al-Turjman, F. (2020). Securing demand response management: A certificate-based access control in smart grid edge computing infrastructure. IEEE Access, 8, 101235-101243.

[93] Cui, G., He, Q., Li, B., Xia, X., Chen, F., Jin, H., ... & Yang, Y. (2021). Efficient verification of edge data integrity in edge computing environment. IEEE Transactions on Services Computing, 15(6), 3233-3244.

[94] Li, B., He, Q., Chen, F., Jin, H., Xiang, Y., & Yang, Y. (2021). Inspecting edge data integrity with aggregate signature in distributed edge computing environment. IEEE Transactions on Cloud Computing, 10(4), 2691-2703.

[95] Figueiredo, R., & Subratie, K. (2020, November). Edgevpn. io: Open-source virtual private network for seamless edge computing with kubernetes. In 2020 IEEE/ACM Symposium on Edge Computing (SEC) (pp. 190-192). IEEE.

[96] Ansari, M. S., Alsamhi, S. H., Qiao, Y., Ye, Y., & Lee, B. (2020). Security of distributed intelligence in edge computing: Threats and countermeasures. The Cloud-to-Thing Continuum: Opportunities and Challenges in Cloud, Fog and Edge Computing, 95-122.

[97] Li, G., Ren, X., Wu, J., Ji, W., Yu, H., Cao, J., & Wang, R. (2021). Blockchain-based mobile edge computing system. Information Sciences, 561, 70-80.

[98] Cheriet, A., & Mekhaznia, T. (2022, October). Secure Mobile Data Offloading in Small Cell Networks. In 2022 4th International Conference on Pattern Analysis and Intelligent Systems (PAIS) (pp. 1-6). IEEE.

[99] Al Omar, A., Rahman, M. S., Basu, A., & Kiyomoto, S. (2017, December). Medibchain: A blockchain based privacy preserving platform for healthcare data. In International conference on security, privacy and anonymity in computation, communication and storage (pp. 534-543). Cham: Springer International Publishing.

[100] Du, L., Zhang, W., Fu, H., Ren, W., & Zhang, X. (2019). An efficient privacy protection scheme for data security in video surveillance. Journal of visual communication and image representation, 59, 347-362. https://doi.org/10.1016/j.jvcir.2019.01.027.

[101] Khan, P. W., Byun, Y. C., & Park, N. (2020). A data verification system for CCTV surveillance cameras using blockchain technology in smart cities. Electronics, 9(3), 484. https://doi.org/10.3390/electronics9030484.

[102] Yallamandhala, P., & Godwin, J. (2022, April). A review on video tampering analysis and digital forensic. In Proceedings of International Conference on Deep Learning, Computing and Intelligence: ICDCI 2021 (pp. 287-294). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-16-5652-1_24.

[103] Jiang, X., Yu, F. R., Song, T., & Leung, V. C. (2020). Intelligent resource allocation for video analytics in blockchain-enabled internet of autonomous vehicles with edge computing. IEEE Internet of Things Journal, 9(16), 14260-14272. Doi: 10.1109/JIOT.2020.3026354.

[104] Barman, N., Deepak, G. C., & Martini, M. G. (2020). Blockchain for video streaming: opportunities, challenges, and open issues. Computer, 53(7), 45-56. Doi: 10.1109/MC.2020.2989051.

[105] Cho, J. R., Kim, H. S., Chae, D. K., & Lim, S. J. (2017). Smart CCTV security service in IoT (internet of things) environment. Journal of digital contents society, 18(6), 1135-1142. https://doi.org/10.9728/dcs.2017.18.6.1135.

[106] Lv, W., Wang, N., Xie, X., & Hong, Z. (2022). A Classification-Based Blockchain Architecture for Smart Home with Hierarchical PoW Mechanism. Buildings, 12(9), 1321. https://doi.org/10.3390/buildings12091321.

[107] Wang, R., Tsai, W. T., He, J., Liu, C., Li, Q., & Deng, E. (2019, February). A video surveillance system based on permissioned blockchains and edge computing. In 2019 IEEE international conference on big data and smart computing (BigComp) (pp. 1-6). IEEE. Doi: 10.1109/BIGCOMP.2019.8679354.

[108] Ottakath, N., & Al-Maadeed, S. A. (2022, September). Reliable Video Forensics Evidence Cataloguing using Video Source device Identification on the Blockchain. In 2022 International Conference on Emerging Trends in Smart Technologies (ICETST) (pp. 1-6). IEEE. Doi: 10.1109/ICETST55735.2022.9922947.

[109] Li, Y., & Wan, Z. (2021). Blockchain-enabled intelligent video caching and transcoding in clustered MEC networks. Security and Communication Networks, 2021. https://doi.org/10.1155/2021/7443260.

[110] Memos, V. A., Psannis, K. E., Ishibashi, Y., Kim, B. G., & Gupta, B. B. (2018). An efficient algorithm for media-based surveillance system (EAMSuS) in IoT smart city framework. Future Generation Computer Systems, 83, 619-628. https://doi.org/10.1016/j.future.2017.04.039

[111] Mercan, S., Cebe, M., Aygun, R. S., Akkaya, K., Toussaint, E., & Danko, D. (2021). Blockchain□based video forensics and integrity verification framework for wireless Internet□of□Things devices. Security and Privacy, 4(2), e143.

[112] Gallo, P., Pongnumkul, S., & Nguyen, U. Q. (2018, June). BlockSee: Blockchain for IoT video surveillance in smart cities. In 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe) (pp. 1-6). IEEE.

[113] Ghimire, S., Choi, J. Y., & Lee, B. (2019). Using blockchain for improved video integrity verification. IEEE Transactions on Multimedia, 22(1), 108-121.

[114] Kumar, V., Singh, A., Kansal, V., & Gaur, M. (2021). A comprehensive survey on passive video forgery detection techniques. Recent Studies on Computational Intelligence, 39-57. https://doi.org/10.1007/978-981-15-8469-5_4.

[115] Latif, S. A., Wen, F. B. X., Iwendi, C., Li-li, F. W., Mohsin, S. M., Han, Z., & Band, S. S. (2022). AI-empowered, blockchain and SDN integrated security architecture for IoT network of cyber physical systems. Computer Communications, 181, 274-283. https://doi.org/10.1016/j.comcom.2021.09.029.

[116] Yazdinejad, A., Parizi, R. M., Dehghantanha, A., Zhang, Q., & Choo, K. K. R. (2020). An energy-efficient SDN controller architecture for IoT networks with blockchain-based security. IEEE Transactions on Services Computing, 13(4), 625-638. Doi: 10.1109/TSC.2020.2966970.

[117] Liu, Y., Yu, F. R., Li, X., Ji, H., & Leung, V. C. (2019). Decentralized resource allocation for video transcoding and delivery in blockchain-based system with mobile edge computing. IEEE Transactions on Vehicular Technology, 68(11), 11169-11185. Doi: 10.1109/TVT.2019.2937351.

[118] Liu, M., Teng, Y., Yu, F. R., Leung, V. C., & Song, M. (2020). A mobile edge computing (MEC)-enabled transcoding framework for blockchain-based video streaming. IEEE Wireless Communications, 27(2), 81-87. Doi: 10.1109/MWC.001.1800332.

[119] Cheriet, A., & Mekhaznia, T. (2022, October). Secure Mobile Data Offloading in Small Cell Networks. In 2022 4th International Conference on Pattern Analysis and Intelligent Systems (PAIS) (pp. 1-6). IEEE. Doi: 10.1109/PAIS56586.2022.9946896.

[120] Yi, S., Qin, Z., & Li, Q. (2015). Security and privacy issues of fog computing: A survey. In Wireless Algorithms, Systems, and Applications: 10th International Conference, WASA 2015, Qufu, China, August 10-12, 2015, Proceedings 10 (pp. 685-695). Springer International Publishing.

[121] Chiang, M., & Zhang, T. (2016). Fog and IoT: An overview of research opportunities. IEEE Internet of things journal, 3(6), 854-864.

[122] Patel, M., Naughton, B., Chan, C., Sprecher, N., Abeta, S., & Neal, A. (2014). Mobile-edge computing introductory technical white paper. White paper, mobile-edge computing (MEC) industry initiative, 29, 854-864.

[123] Ometov, A., Molua, O. L., Komarov, M., & Nurmi, J. (2022). A survey of security in cloud, edge, and fog computing. Sensors, 22(3), 927.

[124] Scott-Hayward, S., Natarajan, S., & Sezer, S. (2015). A survey of security in software defined networks. IEEE Communications Surveys & Tutorials, 18(1), 623-654.

[125] Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., & Sabella, D. (2017). On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. IEEE Communications Surveys & Tutorials, 19(3), 1657-1681.

[126] Novo, O. (2018). Blockchain meets IoT: An architecture for scalable access management in IoT. IEEE internet of things journal, 5(2), 1184-1195.

[127] Paavolainen, S., & Carr, C. (2020). Security properties of light clients on the ethereum blockchain. IEEE Access, 8, 124339-124358.

[128] Benet, J. (2014). Ipfs-content addressed, versioned, p2p file system. arXiv preprint arXiv:1407.3561.

[129] Ayo, F. E., Folorunso, S. O., Abayomi-Alli, A. A., Adekunle, A. O., & Awotunde, J. B. (2020). Network intrusion detection based on deep learning model optimized with rule-based hybrid feature selection. Information Security Journal: A Global Perspective, 29(6), 267-283.

[130] Stafford, V. (2020). Zero trust architecture. NIST special publication, 800(207), 800-207.

[131] Gentry, C. (2009, May). Fully homomorphic encryption using ideal lattices. In Proceedings of the forty-first annual ACM symposium on Theory of computing (pp. 169-178).

[132] Fontes, R. R., Afzal, S., Brito, S. H., Santos, M. A., & Rothenberg, C. E. (2015, November). Mininet-WiFi: Emulating software-defined wireless networks. In 2015 11th International Conference on Network and Service Management (CNSM) (pp. 384-389). IEEE.

[133] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., ... & Yellick, J. (2018, April). Hyperledger fabric: a distributed operating system for permissioned blockchains. In Proceedings of the thirteenth EuroSys conference (pp. 1-15).

[134] Varga, A., & Hornig, R. (2008, March). An overview of the OMNeT++ simulation environment. In Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops (pp. 1-10).

[135] Lee, D., & Park, N. (2021). Blockchain based privacy preserving multimedia intelligent video surveillance using secure Merkle tree. Multimedia Tools and Applications, 80(26), 34517-34534. https://doi.org/10.1007/s11042-020-08776-y.

[136] Sharma, P., Jindal, R., & Borah, M. D. (2023). Blockchain-based distributed application for multimedia system using Hyperledger Fabric. Multimedia Tools and Applications, 1-27. https://doi.org/10.1007/s11042-023-15690-6.

[137] Klein, D., & Jarschel, M. (2013, March). An OpenFlow extension for the OMNeT++ INET framework. In Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques (pp. 322-329).

[138] Viega, J., Messier, M., & Chandra, P. (2002). Network security with openSSL: cryptography for secure communications. ” O’Reilly Media, Inc.”.

[139] Wei, D. A. I. (2004). Crypto++ library 5.1-a free c++ class library of cryptographic schemes. http://www. cryptopp. com/.

[140] Ottakath, N., & Al-Maadeed, S. A. (2022, September). Reliable Video Forensics Evidence Cataloguing using Video Source device Identification on the Blockchain. In 2022 International Conference on Emerging Trends in Smart Technologies (ICETST) (pp. 1-6). IEEE. Doi: 10.1109/ICETST55735.2022.9922947.

[141] Memos, V. A., Psannis, K. E., Ishibashi, Y., Kim, B. G., & Gupta, B. B. (2018). An efficient algorithm for media-based surveillance system (EAMSuS) in IoT smart city framework. Future Generation Computer Systems, 83, 619-628. https://doi.org/10.1016/j.future.2017.04.039