Securing Service Migration in Multi-Access Edge Computing

Pasika Ranaweera

UCD student number: 18204751

The thesis is submitted to University College Dublin
in fulfillment of the requirements for the degree of
Doctor of Philosophy in Computer Science

School of Computer Science

Head of School: Associate Professor Neil Hurley

Research Supervisor: Dr. Anca Jurcut

Research Co-Supervisor: Dr. Madhusanka Liyanage

May 2023

I dedicate this dissertation work to my Family:
my mother Swineetha Weerasinghe, my late father Priyantha Ranaweera, my brother Harshika Ranaweera, and my loving wife Vindya Veragoda....
You have made me stronger, given me confidence, and encouraged me to raise beyond the levels of my ability.
I believe we will have many years of great times together to celebrate this achievement.
The best of luck to you in all of your endeavors!

Acknowledgements

"None of us, including me, ever do great things. But we can all do small things, with great love, and together we can do something wonderful."

Quoting Mother Teresa I wish to acknowledge this dissertation bares the hard work, commitment, and guidance of many collaborators that I wish to declare in this attempt.

First and foremost, my heartiest gratitude goes to my supervisor, Dr. Anca Delia Jurcut, who has taken care of me during the last four and a half years and enriched me with unconditional guidance, motivation, encouragement, and support throughout my Ph.D. career, and during the concluding stages where I was writing my thesis. The research environment she created was ideal for any individual, who started a Ph.D. and expects to perform exceptionally. Her conduct in guiding me was exceptional, as I was given plenty of time to think and solve my problems, and she trusted me and waited for me to perform patiently. The experience I have gathered through working with her on BDIC for teaching and instructing was extremely rewarding for me to perform in an international venue. She was always available for me to solve my doubts and acted in my best interest at all times. Your intelligent, kind, and polite personality and working gestures will be always memorable to me in this stage where I am about to celebrate the biggest achievement of my life. Thank you for the wonderful memories and for giving me the opportunity to be your student. I am wishing you all the best for your future...!!!

I would like to convey my genuine and unconditional gratitude to Dr. Madhusanka Liyanage, my greatest mentor in life. From directing me to find a Ph.D. opportunity to guiding me to complete this milestone in my life, you were there with me to pave the path to where I am today. You are one of the most efficient, knowledgeable, intelligent, and hard-working individuals I have ever

come across in my life. These traits have put a positive attitude to my life. Especially on time management. The opportunities you gave me for research publications, reviewing, project application, and finally my future career direction are extremely valuable for me. Your confidence in attempting for the highest standards when it comes to research journals, conferences, venues, and academic projects has pushed me forward beyond my ability and ushered me to a level I have never thought of reaching before. In fact, you are my inspiration to pursue a career in academia. You deserve all the praise, achievements, and gratitude.

I would like to thank the members of my Doctoral Study Panel: Dr. Mark Scanlon, and Dr. Pavel Gladyshev. Your constructive criticisms have shaped my Ph.D. direction and led me to contribute quality outcomes from my Ph.D. study. My heartiest gratitude to the two members for spending their valuable time assessing my work and guiding me on the correct path. Special thanks to Dr. Mark Scanlon for acting as a referee to my IRC scholarship application.

I am always faithful to my research collaborators Dr. Indika Anuradha Balapuwaduge, Awaneesh Kumar Yadav, Vashish N. Imrith, Dr. Chamitha De Alwis, and Tharindu Gamage. Your contribution and commitment have aided me to complete this dissertation. I hope we could continue our collaboration in the future.

I am forever grateful to Prof. Liam Murphy and Prof. John Murphy for the excellent feedback I received during the presentations on the Performance Engineering Laboratory sessions. Your constructive rationales have raised my fidelity toward the Ph.D. topic and allowed me to change my perspective on important aspects under my research directive.

The Netslab team has always given me valuable guidance and support to engage in collaborative research directives. I am most grateful for that. Further, my colleagues, Dr. Mahmoud Said El Sayed Abdallah, Dr. Eric Gyamfi, Dr. Muhammad Zahid Iqbal, and Dr. Manaz Kaleel; I am forever grateful for

your company, support, and friendship. My years in Dublin would have been dull without your precious acquaintance.

Special thanks are conveyed to my Sri Lankan colleagues in UCD: Ranul, Shalitha, and Rajitha for your continuous friendship.

Special acknowledgment goes to the UCD School of Computer Science for hosting me for the past four and a half years in their care. Your kind and attentive care for your graduate students is a quality that is valued by most. It was an immense opportunity to study in this great citadel of learning.

Last but not least, my cordial gratitude goes to my mother and my brother, who has supported me unconditionally during this time span. Finally, I wish to convey my distinct appreciation to my loving wife, Vindya Veragoda; for standing by me during these extremely dull and stressful days of our lives.

ABSTRACT

Edge computing paradigms were an expedient innovation for elevating the contemporary standards of mobile and Internet networks. Multi-Access Edge Computing (MEC) is an emerging edge computing paradigm that has the potential to overcome the disparity between the prevailing and envisioned networking architectures suited to realize 5G-based applications. As specified in MEC standardization, edge computing serviceable infrastructures are running on virtualization technologies to provide dynamic and flexible service instances to cater to User Equipment (UEs) of various formations to accomplish diverse use cases. Since the inception and operation of the services are executing at the edge level gNodeBs (gNBs), migration of services between gNBs is an imminent occurrence in edge computing that is contriving challenges to its feasible deployment. Security and Service Level Guarantee (SLG) requirements are vital parameters for such service migration operations conducted through gNB-to-gNB (g2g) connecting channels.

With the advent of 5G, local operators are granted the ability to launch services in the mobile network, and such operators are not quite trustable due to the scalability of 5G. There is always a possibility of a fake gNB being launched by an adversary with replicated communication protocols. Further, the g2g Service Migration Channel (SMC) is subjected to Man-in-the-Middle (MitM)

type intrusions that could invoke threat vectors ranging from simple eavesdropping threats to injection of malicious agents to the migrating content. In addition, emerging applications and use cases such as autonomous vehicles and unmanned aerial vehicles are setting a very low service level latency requirement. Therefore, attacks conducted to intentionally impede the services are impacting the MEC's performance substantially. On the contrary, a formidable level of security applied to mitigate such threat vectors can overwhelm the bandwidth of the SMC and aggregate processing latency from cryptographic operations. Therefore, in this Ph.D., 1) holistic security concerns and identity verification among active agents featured by the service migration phenomena of MEC deployments, and 2) optimization of the security level considering the service level latency specified by SLGs, are the prime research problems that are addressed.

A MEC Service Migration Security Framework (MEC-SMSF) is proposed and developed for specifying the methodology to securely migrate a service instance within MEC-enabled gNodeBs. This framework incorporates an authentication protocol called MEC Service Migration Authentication Protocol (MEC-SMAP) to ensure identity verification among the parties involved in a service migration through authentication and to secure the migrating content through a robust g2g channel establishment. The proposed protocol was verified employing both formal and informal methods while feasibility was validated using a test-bed prototype environment.

In addition, MEC-SMSF embeds a model for optimizing the level of security applied for migrating content based on the instantaneous bandwidth utilization of the channel: called MEC Service Migration Security Management (MEC-SMSM), which guarantees the satisfaction of SLGs. The proposed model and its standardization benchmarks classify the distinct security mechanisms em-

ployed for holistic security solutions based on their derived security cost. A Markovian model is proposed to formalize an estimation scheme, which is predicting the most probable security setting through probabilistic means. The proposed models and their concepts are validated with simulations and the prototype implementation of the MEC-SMSF verified the feasibility of this solution.

The proposed and validated solutions in this thesis guarantee that MEC-based service migrations to be conducted securely, efficiently, and reliably to maintain the service continuity of high-priority critical applications envisaged in the future.

Keywords : MEC, Service Migration, Security, Authentication, Security Optimization

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Statement of Original Authorship

I hereby certify that the submitted work is my own work, was completed while registered as a candidate for the degree stated on the Title Page, and I have not obtained a degree elsewhere on the basis of the research presented in this submitted work.

16/05/2023

Pasika Ranaweera
(Student Number: 18204751)

Date

ACRONYMS

3GPP	Third Generation Partnership Project
4G	Fourth Generation Telecommunication Networks
5G	Fifth Generation Telecommunication Networks
AI	Artificial Intelligence
AR	Augmented Reality
AV	Autonomous Vehicle
BLE	Bluetooth Low Energy
BS	Base Station
CC	Cloud Computing
CDN	Content Delivery Network
CFS	Customer Facing Service
CIA	Confidentiality, Integrity, and Availability
CPS	Cyber-Physical System
D2D	Device-to-Device
DDoS	Distributed Denial of Service
DoS	Denial of Service
E2E	End-to-end
eMBB	enhance Mobile Broadband

eNodeB	Evolved Node B
ETSI	European Telecommunications Standards Institute
GDPR	General Data Protection Regulation
GSM	Global System for Mobile Communication
GT	Game Theory
ICN	Information-Centric Networking
IDS	Intrusion Detection Scheme
IIoT	Industrial Internet of Things
IoT	Internet of Things
ISG	Industry Specification Group
ITS	Intelligent Transport System
LAN	Local Area Network
LADN	Local Area Data Network
LPWAN	Low-power Wide Area Network
LTE	Long Term Evolution
M2M	Machine-to-Machine
MANET	Mobile Ad-hoc Network
MANO	Management and Network Orchestration
MCC	Mobile Cloud Computing
ME	Mobile Edge
MEC	Multi-Access Edge Computing
MEH	Mobile Edge Host
MEO	Mobile Edge Orchestrator
MEN	Mobile Edge Network
MEPM	Mobile Edge Platform Manager
MES	Mobile Edge Service
MitM	Man-in-the-Middle

mmWave	millimeter-Wave
MNO	Mobile Network Operator
MR	Mixed Reality
MTC	Machine Type Communication
NB-IoT	Narrow-band IoT
NFC	Near Field Communication
NFV	Network Function Virtualization
NS	Network Slicing
OSS	Operation Support System
PbD	Privacy by Design
QoE	Quality of Experience
RAN	Radio Access Networks
RFID	Radio-Frequency Identification
SDN	Software-Defined Networking
SDP	Software-Defined Privacy
TV	Threat Vector
UALCMP	User Application Life-Cycle Management Proxy
UAV	Unmanned Aerial Vehicles
UE	User Equipment
UHD	Ultra High Definition
URLLC	Ultra-reliable Low-latency Communication
V2E	Vehicle to Everything
V2I	Vehicle to Infrastructure
VIM	Virtualization Infrastructure Manager
VM	Virtual Machine
VNF	Virtual Network Function
VR	Virtual Reality

WAN	Wide Area Networking
WLAN	Wireless Local Area Network

CHAPTER

ONE

INTRODUCTION

Moore's Law suggests the processor speed is exponentially incrementing over time [1, 2]. Hence, the number of Internet of Things (IoT) devices employed in industries serving Big Data applications are thriving with the possibility of proliferated processing capability in miniaturized devices. Moreover, improved smart device usage literacy of the general public in the modern era is enabling social Internet platforms to launch cumbersome bandwidth-consuming applications for elevating their subscriptions with immersive Quality of Service (QoS). It is estimated that the number of mobile terminals is reaching 2.8 billion by 2019 and monthly mobile data traffic is reaching beyond 49 exabytes by 2021 according to Cisco [3]. Thus, deployments of billions of smart devices demand access capacity and bandwidth requirements from the access interfaces of mobile Base Stations (BSs).

The fifth-generation (5G) mobile technology is the seminal advancement explored by Mobile Network Operators (MNOs) to reach beyond the constrictions of the prevailing network architecture. To achieve the novel requirements of enhanced performance, portability, interoperability, elasticity, reliability, spectral, and energy efficiency; a network softwarization approach should be followed by the evolving mobile networks [4]. Virtualization, service migration, orchestration, and service automation (as in service function chaining [5]) are the main phases of paving the path towards 5G and beyond 5G mobile

paradigms[6]. As the core and backhaul portions of the emerging mobile networks are softwarized; techniques of ultra-dense networks, massive Multiple-Input-Multiple-Output (MIMO), and high-frequency communication are prominent methods for improving the wireless access network [3]. Due to these technological improvements, 5G guarantees a 1000 times enhancement of the capacity than its predecessor. The guaranteed performance metrics of 5G are: data rates up to 10 Gb/s, service level latency below 1 ms, ultra-high reliability of 99.99999%, reduced energy consumption of 90%, and support for 300,000 devices within a single cell [7, 8].

Even with the softwarized 5G core network, facilitating the diverse requirements demanded by the IoT-based devices is still a predicament due to the drawbacks of existing service provisioning infrastructure [9]. A typical intelligent, autonomous application or service executed on a smart IoT device requires connectivity to the centralized cloud services for circulating control information and authentication credentials in case of an authorization mechanism. This connectivity is generally linked through the Internet for facilitating a communication channel with strong cryptic credentials. This ubiquitous and bandwidth-consuming connectivity to out-of-proximity entities is restricting the responsiveness of the applications, hindering the real-time services guaranteed by forthcoming mobile technologies. Conventional cloud computing architecture fails to provide such emerging real-time services [10]. The geographically distant placement of data centers and limited access capacity contrives unintended delays and jitters that compromise the entire service infrastructure. Moreover, cloud servers are incapable of servicing billions of IoT devices ubiquitously. These limitations in the cloud computing paradigm enforce vulnerabilities that can be exploitable by adversaries [11]. Moreover, privacy is a major concern with the outsourcing-based cloud computing service models [12]. Most cloud service providers are violating the locational and data privacy of their consumers.

1.1 The Concept of Edge Computing

In order to overcome these constrictions in storage and processing service models, Edge Computing (EC) as a paradigm was introduced in the 1990s with Content Delivery Networks (CDN) that decentralized the data center functions [13]. The main objective of EC was to extend the functions offered from cloud computing to the edge of the mobile network [7]. With in-proximity dispensing of cloud functions at the edge, drawbacks of the cloud paradigm could be mitigated. In fact, this architectural paradigm shift is the raison d'etre for 5G and beyond 5G-based concepts to achieve guaranteed performance metrics. There are various flavors of edge concepts introduced for expanding this notion. Multi-Access Edge Computing (MEC), Fog computing, Mobile Cloud Computing (MCC), Cloudlets, and Transparent Computing (TC) are such directives followed by research communities [7, 3]. Out of these concepts, however, MEC and fog computing are leading to be adopted pragmatically and in terms of standardization. In this thesis, we are investigating the MEC paradigm as its standardization is much more convincing than the other concepts.

MEC is a nascent paradigm proposed by the European Telecommunications Standards Institute (ETSI) to overcome the intricacies of highly evolving mobile and wireless communication networks. The underlying principle of MEC is to extend the Cloud Computing (CC) capabilities to the edge of the mobile network to curtail the attributed constraints on existing cloud infrastructure [7]. More anecdotally, MEC complements the corporate data and processing centers provisioning to compute, storage, networking, and data analytic resources at locations in proximity to the data source as illustrated in Fig. 1.1 [14]. MEC is standardized as an initiative to achieve the granted 5G specifications stated above. In order to attain these requirements, migrating the service infrastructure to a proximate location to the User Equipment (UE) is a critically intrinsic approach. Thus, the MEC paradigm is formed and designed with the above intentions. Deployment of MEC systems is forcing unconventional architectural alterations to the current cloud-native services, where unintentional vulnerabilities are imminent; and exploring such flows would raise the feasibility of the technology through thorough assimilation. In addition, the completely

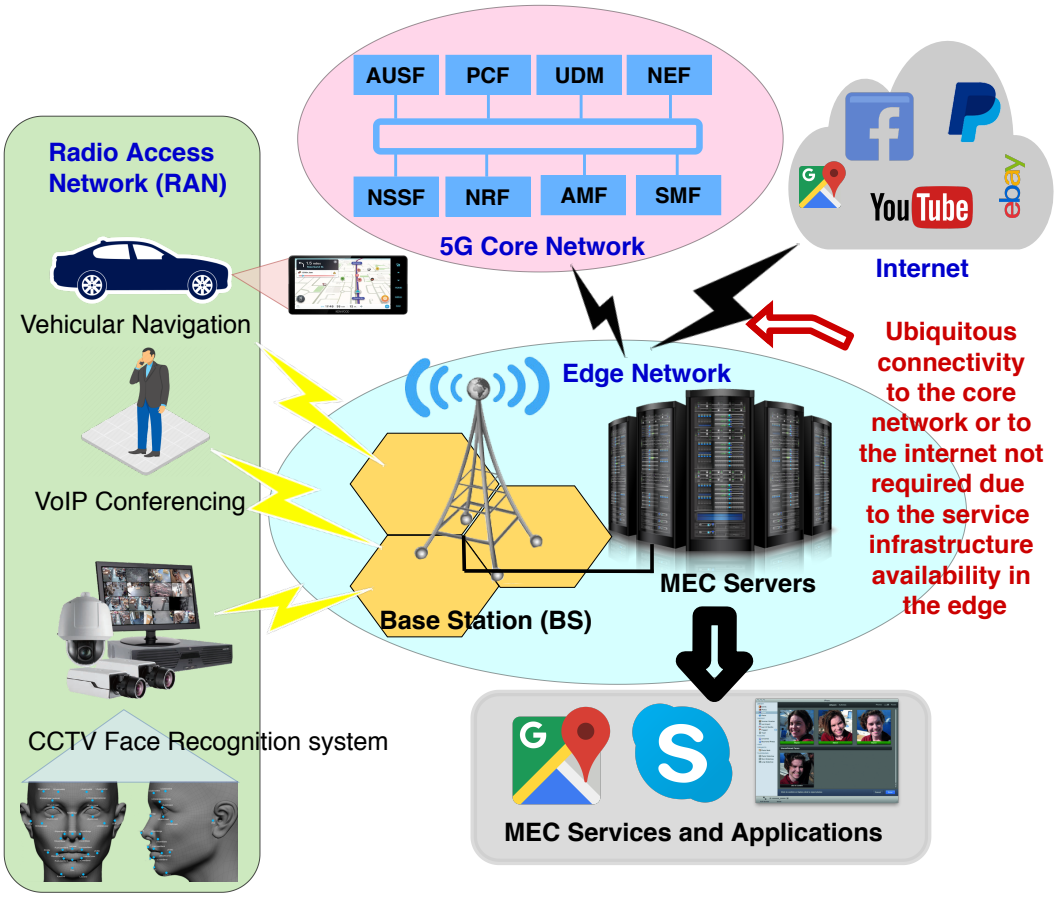


Figure 1.1: What is MEC

autonomous serviceable infrastructure demanded by the MEC systems deems the utilization of virtualization technologies that feature autonomous orchestration [15]. Embedding virtualization technologies are contriving novel issues and challenges. Such issues were sought out as a form of threat vectors within a pragmatic MEC deployment at the initial stages of this Ph.D. (stated in Section 2.2.3). This study, published in [16] revealed the architectural threat vectors of network slicing, traffic steering, service migration, and mobility management. Out of these threat vectors, service migration is an area that wasn't approached through research from the security perspective.

In the existing CC service architecture depicted in Fig. 1.2, all the emanated service requests in the Radio Access Network (RAN) are traversed to the cloud servers, which are located at different global locations due to

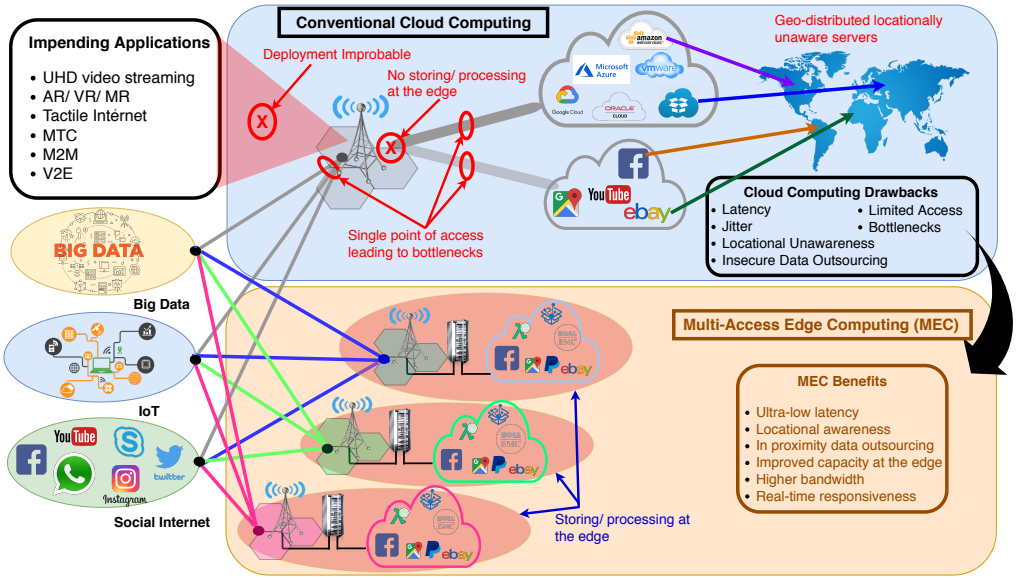


Figure 1.2: MEC Paradigm and its requirement

the non-existent storage and processing platform at the BS. The subscribers are unaware of the exact locations of the servers due to the outsourcing process. This fact is raising security and privacy concerns, as the personal data of the subscribers are handled by third parties without any concrete assurances. The channel conveying the elevated service requests and data to the cloud servers is bound to form a bottleneck in the network traffic in addition to the RAN access interface [17]. Therefore, CC-based services are expected to endure latency issues, jitter, and unresponsiveness in addition to the security ramifications from service interruption–based attacks perpetrated by adversaries. These factors prove the improbability of successfully deploying impending applications with 5G technology such as Ultra High Definition (UHD) video streaming, Augmented Reality (AR), Virtual Reality (VR), Mixed Reality (MR), Tactile internet, Machine Type Communication (MTC), Machine-to-Machine (M2M), Unmanned Aerial Vehicle (UAV), and Vehicle-to-Everything (V2E). The storage and processing infrastructure facilitated by MEC deployments, however, are ensuring the benefits of ultra-low latency, locational awareness, proximate data outsourcing, and improved capacity in edge devices. These features enable higher bandwidth and real-time respon-

siveness to the subscriber applications. Moreover, MEC-based services within the RAN enhance computational processing power to avoid bottlenecks with directed mobile traffic [18]. These factors are making MEC the preeminent technology behind 5G deployment.

1.1.1 Edge Computing Associated Service Migration Process

As stated above, the emergence of edge computing paradigms has introduced the concept of service migration to cater to the myriads of IoT devices for heterogeneous and ubiquitous connectivity with its EC infrastructure to maintain service continuity. The MEC-based services are offered from the nearest MEC-enabled BS or the gNodeB (or gNB) to the subscriber UE. Since the service instance or the program executing at the edge platform is originating at the serving gNB, other MEC-enabled gNBs don't possess the service instance with the same operational configuration. In a situation where the subscriber is traversing beyond the range of the currently serving MEC gNB, the service instance should be migrated to a gNB with MEC capabilities, that is located in proximity to the subscriber roamed location. Once migrated and configured to the roamed MEC infrastructure, offered service to the consumer continues through the communication channels of the roamed BS. The QoS and Quality of Experience (QoE) aspects of the offered MEC-based service are entirely dependent on the seamless operation of the migration process. Latency or a delay caused in the migration process will result in disruption of the service to the consumer device; thereby impacting both QoS and QoE factors negatively. Thus, service migration within edge computing platforms is a weaker aspect of MEC that forecasts inevitable issues.

1.1.2 Use Case

In order to emphasize the importance of the migration delay in MEC-based systems, I have considered the emerging application of autonomous/driverless vehicles as a use case, which is depicted in Fig. 1.3. The complete

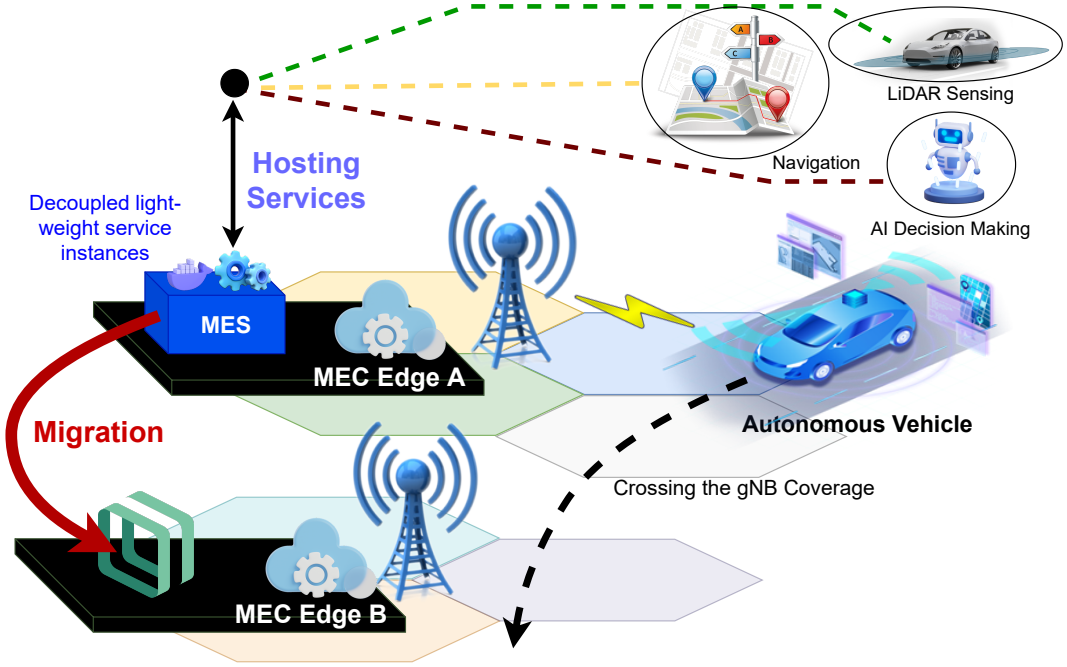


Figure 1.3: Autonomous Vehicles Use Case

United States map is taking around 41 TB of space to be stored in a digital memory [19]. An isolated and disconnected vehicle without any connection to the Internet is required to store the entire map of its origin for non-assisted autonomous driving. Further, real-time updates of weather conditions, traffic and navigation information are critical for achieving a smoothly driven Autonomous Vehicle (AV). AVs are operating based on their sensory acquisitions gathered from cameras and various sensors or actuators (LiDAR, Sonar, Radar, ultrasonic. . . etc.) embedded in them. For accurate and seamless processing, several Graphic Processing Units (GPUs) should be operated in the AV; while compromising the battery of the AV [20].

Embedding a powerful processor with super-computing capabilities would be highly expensive and reduces the feasibility of realizing the AV concept from the marketing perspective. Moreover, advanced navigation functions, rapid sensory acquisition of data, and execution of autonomous algorithms or AI-based decision-making processes consume the available processing power of the AV controlling unit. The battery and fuel consumption would be adverse

to cater to these processing requirements. Thus, offloading such intensive tasks to an edge infrastructure that guarantees the End-to-End (E2E) delay of 100 ms is improving the odds of realizing this concept. In addition, sensory inputs drawn by multiple vessels and Road Side Infrastructure (RSI) devices can be used to deduce accurate results for decision-making through MEC-based edge infrastructure. Thus, for realizing AVs, edge computing is an evident deployment scenario due to these reasons.

1.2 Problem Statement

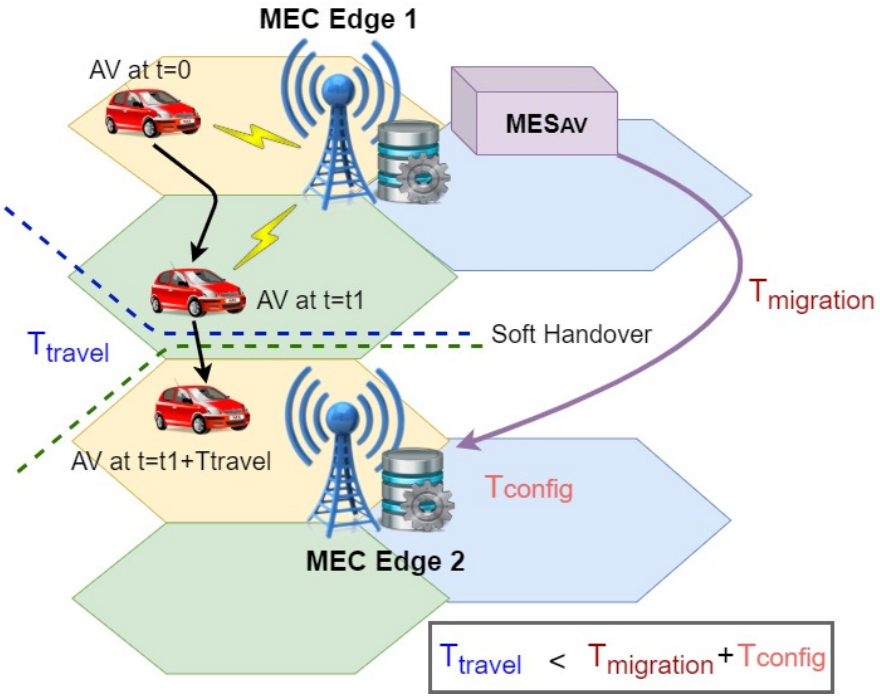


Figure 1.4: Latency of service migration causing service disruption

The latency associated with the link between the edge infrastructure and the AV is imperative for service continuity. In a handover situation where the AV is shifting from one serving gNB to another, the communication channel latency would not be affected due to the existence of soft-handover scenarios. However, in a MEC deployment, AV-based services are not available in all the

gNBs. In such a scenario, AV service should be migrated to the roamed gNB for maintaining service continuity. A typical service migration takes around 1 to 5 seconds or more for service instances [21]. If the service migration process is initiated at the handover commencing instant, the migration and configuration (installing and configuring the service instance in the novel gNB environment) latency is exceeding the time taken for the handover process as depicted in Fig. 1.4. Thus, the delay associated with the migration process is a factor that should be contemplated before realizing low-latency-prone applications.

Migration channels are prone to Man-in-the-Middle (MitM) type attacks perpetrated as malicious code injection, Relay, or advanced persistent threats. Instilling data or code fragments into the migrating content is probable in the current edge computing context as the links between the gNBs are established via the air interface. This type of insertion would be impossible to detect once the entire migrated content is configured for operation at the roamed gNB. As the services are migrated as Virtual Machines (VMs), VM-based attack scenarios such as privilege escalation, VM manipulation, VNF location shift, and VM software vulnerabilities are probable for intercepting the migration channel between the MEC edge platforms (i.e. between gNBs). In addition, service-impeding attempts committed to cause Denial of Service (DoS) type attacks can impact these latency-prone services severely.

There is a clear relation between security and service continuity measures of service migration. Security measures cannot be employed with a high level of encryption and tunneling-based approaches due to time criticalness. Moreover, current service migrations are VM based; where the size of a VM is several GBs. In the context of migration, this is cumbersome. In order to solve the latency issue associated with the migration, pre-migration of the service prior to initiation of the handover sequence or alleviating the cumbersome content transferring among the MEC gNBs can be identified. In this thesis, lightweight virtualization-based service instances are employed instead of hypervisor-based systems (VMs) for migration. The significance of the lightweight approach or containerization is its lesser size and flexible nature compared to VMs. Containers inherit dynamicity, flexibility, compatibility,

and auto-configurable capabilities. Thus, minimizing the delay is possible with this solution. However, containers are attributing vulnerabilities of their own in the security context. Therefore, in this research, the main focuses are:

1. Securing the service migration channel between the serving gNB and the roaming gNB during the pre-migration and migration stages.
2. Exploiting the relationship between security and latency of MEC-based edge-to-edge migrations utilizing lightweight virtualization.

As stated above, the main intention of this research is to investigate the security issues of MEC-based service migration processes considering time as a critical factor. Though, when designing security mechanisms to mitigate possible threats; factors such as communication and processing resource availability at the edge infrastructure determine the level of security that can be offered. Moreover, defining security requirements should be specific to the context of MEC-based service migrations.

Thus, the overall problem statement of my research can be stated as:

"How to improve security and efficiency of the service migration process between MEC edge systems utilizing lightweight virtualization?"

1.3 Research Questions

Four Research Questions (RQs) are stated in this thesis. The hypothesis for each RQ is declared along with it. The four defined RQs and corresponding hypotheses stated are quantifiable either qualitatively or quantitatively. In order to solve these RQs, I have formalized Research Objectives (ROs) under each RQ. The defined ROs under each RQ characterize the general targets specified to solve the overall RQs. All the contributions of this research are targeted to accomplish the stated ROs. The research questions of this study can be stated as follows:

1.3.1 Research Question 1

R1 - *What are the critical security threats and vulnerabilities in MEC?*

Hypothesis: It is possible to identify threat vectors of an actual envisaged MEC deployment scenario (according to the ETSI standards) and thereby determine the plausible attack vectors.

Research Objectives:

RO 1.1 Investigate security threats and vulnerabilities in a MEC deployment.

RO 1.2 Select a critical and nuanced issue out of the identified threats.

1.3.2 Research Question 2

R2 - *What are the security requirements for migration within MEC edge systems considering latency, communication, and resource availability aspects?*

Hypothesis: Security requirements can be defined for the possible MitM-based attack vectors perpetrated at the migration channel to transfer lightweight virtualized containers taking resources of the MEC edge infrastructure into account. Further, security levels can be specified where each level is effectively assigning developing security mechanisms on the migration channel based on the available resources at the edge infrastructure.

Research Objectives:

RO 2.1 Identify the communication and processing resource requirements of the MEC edge infrastructure that impact latency and security in service migration scenarios.

RO 2.2 Select suitable technologies for the expected migration model that has provisions for alleviating latency, enforcing security measures, and are capable of optimizing the security level.

RO 2.3 Determine the impact of MitM-type attacks perpetrated on the service migration channels.

RO 2.4 Define security goals to secure the service migration channel with CIA assurance and factors impacting the latency.

1.3.3 Research Question 3

R3 - ***How to develop a MEC-based service migration security framework that satisfies the identified requirements?***

Hypothesis: A security framework can be developed based on the specified requirements (identified in step R2). It is possible to convert the defined security requirements into a logical form (i.e. the identified requirements are formalized into logic rules that can be verified if the desired goals are achieved and that the designed security mechanisms are sound). All the developed security mechanisms can be amalgamated into a holistic framework that migrates containerized service instances.

Research Objectives:

RO 3.1 Identify the security mechanisms that can address the security goals defined in RO 2.4.

RO 3.2 Propose a security protocol for MEC-based service migrations integrating identified security mechanisms.

RO 3.3 Design and develop a MEC platform as a framework suited for performing service migrations from the identified technologies from RO 2.1.

RO 3.4 Propose a model to optimize the relationship between security and latency incorporating the requirements identified in RO 2.1.

1.3.4 Research Question 4

R4 - *How to evaluate and benchmark the proposed solution in a MEC edge platform?*

Hypothesis: Formal verification methods can be used to verify the proposed security mechanisms. The parameters to be defined for evaluating the framework benchmark the performance metrics of all the edge-to-edge service migration processes. The evaluated latency of the migration process of the developed framework is not affecting the service quality.

Research Objectives:

RO 4.1 Verify the proposed security protocol formally and informally while validating its feasibility through performance metrics.

RO 4.2 Measure and benchmark the performance of the developed MEC service migration security framework.

RO 4.3 Evaluate the validity of the proposed security optimization model leveraging the benchmarked parameters of RO 4.2.

1.4 List of Publications

This section lists the completed publications during my Ph.D. study that completely or partially contributed to formalizing this Ph.D. thesis. In addition, the publications still in the submission or revision stages are also presented below. The list of the publications in this dissertation is arranged according to the target venue (i.e. Journal, Magazine, Conference, Poster, Demo, or Book chapter) and is not based on the date of publication or any other order.

Peer-Reviewed Journal Papers

[16] P. Ranaweera, A. D. Jurcut, and M. Liyanage, "Survey on Multi-Access Edge Computing Security and Privacy," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1078–1124, 2021

[22] P. Ranaweera, A. Jurcut, and M. Liyanage, "MEC-enabled 5G Use Cases: A Survey on Security Vulnerabilities and Countermeasures," *ACM Computing Surveys (CSUR)*, vol. 54, no. 9, pp. 1–37, 2021

[23] A. Jurcut, T. Niculcea, P. Ranaweera, and N.-A. Le-Khac, "Security Considerations for Internet of Things: A Survey," *SN Computer Science*, vol. 1, no. 4, pp. 1–19, 2020

Magazine Papers

[24] P. Ranaweera, C. de Alwis, A. D. Jurcut, and M. Liyanage, "Realizing Contact-less Applications with Multi-Access Edge Computing," *ICT Express*, vol. 8, no. 4, pp. 575–587, 2022

[17] P. Ranaweera, M. Liyanage, and A. D. Jurcut, "Novel MEC based Approaches for Smart Hospitals to Combat COVID-19 Pandemic," *IEEE Consumer Electronics Magazine*, vol. 10, no. 2, pp. 80–91, 2020

Peer-Reviewed Conference Papers

[25] P. Ranaweera, A. D. Jurcut, and M. Liyanage, "Realizing Multi-Access Edge Computing Feasibility: Security Perspective," in *2019 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2019, pp. 1–7

[15] P. Ranaweera, V. N. Imrith, M. Liyanage, and A. D. Jurcut, "Security as a Service Platform Leveraging Multi-Access Edge Computing Infrastructure Provisions," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6

[26] P. Ranaweera, A. Jurcut, and M. Liyanage, "Service Migration Authentication Protocol for MEC," in *2022 IEEE Global Communications (GLOBECOM) Conference*. IEEE, 2022

[27] V. N. Imrith, P. Ranaweera, R. A. Jugurnauth, and M. Liyanage, "Dynamic Orchestration of Security Services at Fog Nodes for 5G IoT," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6

[28] V. N. Imrith, P. Ranaweera, S. Damree, and M. Liyanage, "Enabling Fog Computing based Dynamic Security Service Function Chaining for 5G IoT," in *2021 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, 2021, pp. 149–154

[29] G. Dilanka, L. Viranga, R. Pamudith, T. D. Gamage, P. Ranaweera, I. A. Balapuwaduge, and M. Liyanage, "A Novel Server Selection Strategy for Multi-Access Edge Computing," in *2021 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, 2021, pp. 414–419

[30] G. Dilanka, L. Viranga, R. Pamudith, T. Gamage, P. Ranaweera, I. A. Balapuwaduge, and M. Liyanage, "A Novel Request Handler Algorithm for Multi-Access Edge Computing Platforms in 5G," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2022, pp. 126–131

International Poster/Demo Papers

[31] P. Ranaweera, A. D. Jurcut, and M. Liyanage, "Identifying Factors Enabling the Enhancement of Service Migration of Multi-Access Edge Computing," in *2022 Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit)*. IEEE, 2022

[32] G. Dilanka, L. Viranga, R. Pamudith, T. D. Gamage, P. Ranaweera, I. A. Balapuwaduge, and M. Liyanage, "MEC-RHA: Demonstration of Novel Service Request Handling Algorithm for MEC," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2022, pp. 1–3

Publications Under Revision

[P1] **(In Revision)** Pasika Ranaweera, Awaneesh Kumar Yadav, Madhusanka Liyanage, and Anca Delia Jurcut. "A Novel Authentication Protocol for 5G gNodeBs in Service Migration Scenarios of MEC." In *IEEE Transactions on Dependable and Secure Computing*.

[P2] **(In Revision)** Pasika Ranaweera, Indika A. M. Balapuwaduge, Madhusanka Liyanage, and Anca Delia Jurcut. "Service Migration Security Framework for Multi-Access Edge Computing." In *IEEE Transactions on Intelligent Transportation Systems*.

Book Chapters

[33] A. D. Jurcut, P. Ranaweera, and L. Xu, “Introduction to IoT Security,” *IoT security: advances in authentication*, pp. 27–64, 2020

[34] E. H. Jayatunga, P. S. Ranaweera, and I. A. M. Balapuwaduge, “Blockchain Advances and Security Practices in WSN, CRN, SDN, Opportunistic Mobile Networks, Delay Tolerant Networks,” in *Revolutionary Applications of Blockchain-Enabled Privacy and Access Control*. IGI Global, 2021, pp. 1–34

1.5 Disseminated Ph.D. Contributions

Out of the 18 publications stated in the Section 1.4, I have selected the publications (some are in the submission/revision stages) of [16], [22], [25], [15], [26], [31], P1, and P2 as the most contributing publications for this thesis. In this section, the selected eight publications are briefly summarized and presented as the main contributions of the Ph.D. The papers together form a coherent working theme, intended to solve the service migration-related security issues addressed in this thesis.

[25]	***Realizing Multi-Access Edge Computing Feasibility: Security Perspective***
Conference	*IEEE Conference on Standards for Communications and Networking (CSCN)*

This paper has presented the current status of the MEC paradigm from a security perspective, especially focusing on its feasibility for deployment. Security is in fact a critical factor for realizing the potential of the MEC for a feasible deployment. Seven locational threat vectors were identified in a MEC deployment scenario. Then attack vectors corresponding to those threat vectors were also revealed, and existing security solutions were mapped with them. Next, state-of-the-art security mechanisms were explicated in the context of applying them to MEC in its design stages. The main intention was to initiate a discussion on security concerns of the MEC paradigm with this research directive. This paper published the findings of the initial investigations of this Ph.D. and the content related to it can be found in Section 3.1.2. As this was a conference paper, positive feedback was received regarding the initiative and several collaboration opportunities were opened up.

[22]	***MEC-enabled 5G Use Cases: A Survey on Security Vulnerabilities and Countermeasures***
Journal	*ACM Computing Surveys (CSUR)*

It is important to understand the potential of the MEC towards the 5G-based emerging technologies to assimilate the actual layout of a MEC deployment. The concepts of automated driving, augmented reality, and machine-type communication are quite sophisticated; and require an elevation of the current mobile infrastructure for launching. The 5G mobile technology serves as the solution; though lacks a proximate networking infrastructure to satisfy the service guarantees. MEC envisages such an edge computing platform. In this survey, security vulnerabilities of key 5G-based use cases deployed in the MEC context were revealed. Probable security flows of each case are specified, while countermeasures are proposed for mitigating them. The standardized MEC architecture has aided to specify the flaws unique to each use case. Novel security solutions that are proposed for cyber-physical systems,

ICN, NFV, and other impending technologies are mapped for each use case in the context of MEC. The excessive discussion on assimilated facts and future directives are reinforcing the presented proposals with comprehension. The contribution of this survey paper completes the investigation related to MEC security in relation to this Ph.D. study.

[16]	***Survey on Multi-Access Edge Computing Security and Privacy***
Journal	*IEEE Communications Surveys & Tutorials*

The ETSI has introduced the paradigm of MEC to enable efficient and fast data processing in mobile networks. Among other technological requirements, security and privacy are significant factors in the realization of MEC deployments. In this paper, the security and privacy of the MEC system are analyzed thoroughly. This work is an extension of the paper [25], where more threat vector types were explored and presented with tutorial content covering all the basics of MEC technology. Locational threat vectors of the ETSI standardized MEC architecture, architectural threat vectors, and other threat vectors relating to background technologies were identified and analyzed. Furthermore, vulnerabilities leading to the identified threat vectors were assimilated and potential security solutions to overcome these vulnerabilities were proposed from the existing literature. The privacy issues of MEC were also highlighted, and clear objectives for preserving privacy were defined. Finally, future directives to enhance the security and privacy of MEC services were presented. The corresponding content related to this research can be found in Sections 2.1 and 3.1. In this paper, service migration was identified as a critical architectural threat vector for MEC, and the shortage of research currently targeting the security of the concept has led me to select service migration as the aspect to focus on in this thesis.

[31]	***Identifying Factors Enabling the Enhancement of Service Migration of Multi-Access Edge Computing***
Conference	*Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit)*

Migration of services within the edge computing nodes/ base stations is an imminent aspect of the envisaged paradigm that has created a lot of attention. The selection of the optimum edge node to migrate the service is such an issue that restricts the advancement of edge paradigms. The sole focus of this research was to identify and validate the factors enabling the optimal migration decision considering the MEC paradigm. This study was conducted to identify the enabling parameters that govern the service migration process of MEC. The presented validation proves the effectiveness of the identified parameters defined in terms of communication, the operational capability of edge nodes, and the backhaul capacity of the migrating channels. These factors will provide a baseline for formulating solutions to current issues of service migration: security, latency, mobility, and handover management. Section 4.2 presents the findings related to this research and the identified factors are used for various developments of the proposed MEC-SMSF.

[15]	***Security as a Service Platform Leveraging Multi-Access Edge Computing Infrastructure Provisions***
Conference	*IEEE International Conference on Communications (ICC)*

This research was conducted as a starting point for developing a viable MEC-based edge computing platform and understanding the dynamics of an autonomous serviceable infrastructure. The service to be deployed was selected as a security service that is hosted from an outsourced serviceable platform, such as MEC. Thus, the main goal of this paper was to prove the feasibility of the Security as a Service (SECaaS) model in the MEC (Multi-access Edge Computing) paradigm. It allows the launching of multiple security services simultaneously at the edge of the network by employing virtualization technologies. The MEC architecture was followed to form the edge platform and launched multiple security services at the edge successfully. The proposed architecture has the ability to dynamically optimize security services by

adapting resources and also adding or removing additional instances of security services to accommodate the dynamic traffic profiles. The methodologies and techniques adapted would be valuable for telecommunication, cloud, and security service providers to enhance their service models in order to cater to an extended consumer base with improved and guaranteed quality. In terms of the contribution to the thesis (mentioned in Section 2.2 and Chapter 7), the technologies sought out and the design I came up with was instrumental in developing the MEC edge computing platform for the MEC-SMSF.

[26]	***Service Migration Authentication Protocol for MEC***
Conference	*IEEE Global Communications (GLOBECOM) Conference*

This paper introduces MEC Service Migration Authentication Protocol (MEC-SMAP), a protocol that takes place prior to the migration initiation and is specifically defined for MEC. This protocol follows the same design of the MEC-SMAP defined in this thesis, but with lesser functions to comply with the scope of the conference paper. The proposed protocol ensures the secure transfer of session key generation parameters through the insecure service migration channel to form a secure channel while ensuring perfect forward secrecy. It introduces an identity verification mechanism through a trusted third-party service. The proposed protocol has been validated through formal analysis using GNY logic and the Scyther tool. Further, a prototype virtualized MEC environment was created to evaluate its feasibility and the impact of the employed security mechanisms. This prototype was leveraged to extend the protocol further in the later stages of the Ph.D. In contrast to paper P1, this protocol lacks the phases of verifying the MES or ME App with the MEC system level. The content related to this publication is partially presented in Chapter 5, as the contribution has been extended.

P1	***A Novel Authentication Protocol for 5G gNodeBs in Service Migration Scenarios of MEC***
Journal	*IEEE Transactions on Dependable and Secure Computing*

As specified in MEC standardization, edge computing serviceable infrastructures are running on virtualization technologies to provide dynamic and flexible service instances to cater to User Equipment (UE) of various formations to accomplish diverse use cases. Since the inception and operation of the services are executing at the edge level gNodeBs (gNBs), migration of services between gNBs is an imminent occurrence in edge computing that is contriving challenges to its feasible deployment. Security and service level latency requirements are vital parameters for such service migration operations conducted through gNB to gNB (g2g) connecting channels. In this paper, the focus is to ensure identity verification among the parties involved in a service migration through authentication and to secure the migrating content through a robust g2g channel establishment. The proposed authentication protocol, MEC-SMAP was designed in accordance with the MEC architectural standardization. The proposed protocol was verified employing four different formal verification techniques: Scyther and AVISPA verification tools; GNY and ROR logical approaches. Further, the proposed protocol was developed in a test-bed environment emulating the MEC system, with an integrated 5G Core network. The content related to this publication can be found in Chapter 5. As specified earlier, this work is an extension of the paper [26], and completes the MEC-SMAP protocol intended by this thesis.

P2	***Service Migration Security Framework for Multi-Access Edge Computing***
Journal	*IEEE Transactions on Intelligent Transportation Systems*

The inverse relationship between the level of applied security and its ensued latency due to aggregated processing times and overhead is a vital research area for emerging technologies. Following the MEC-enabled Autonomous Vehicles as the impending use case, this paper exploits the stated inverse relationship via a framework proposed to ensure secure service migrations between gNBs. In order to standardize and formulate the applied

security in the context of mobile networks, the term Security Profile was introduced and its applicability and formation were described extensively. In this research, the intention is to propose a service migration security framework (i.e. MEC-SMSF), that specifies the methodology to securely migrate a service instance within MEC-enabled gNodeBs. In addition, this framework embeds a model for optimizing the level of security applied for the migrating content based on the instantaneous bandwidth utilization of the channel, that maintains the SLGs. Further, the proposed model and its concepts are validated with simulations and a prototype implementation. This paper introduces the concept of the MEC-SMSF and augments the concept by modeling the formation of the security profile, mechanism of applying the security, security cost valuation, and optimized security profile selection based on the instantaneous BW of the channel (i.e. security management). The MEC-SMSF prototype utilized in this research is the same testing environment demonstrated in this thesis (i.e. in Chapter 6).

Table 1.1: Research Publications: Covered Research Objectives and Their Relevance to the Thesis Content

P. No.	Related Chapter/ Sections	Research Objectives (ROs)												
		1.1	1.2	2.1	2.2	2.3	2.4	3.1	3.2	3.3	3.4	4.1	4.2	4.3
[25]	3.1.3	✓												
[22]	2.1,3.1	✓												
[16]	2.1,3.1	✓	✓											
[31]	4.2			✓										
[15]	2.2, Ch 7				✓									
[26]	Ch. 5					✓	✓	✓	✓	✓		✓		
P1	Ch. 5					✓	✓	✓	✓	✓		✓		
P2	Ch. 6									✓	✓		✓	✓

Table 1.1 specifies the relation of each publication towards this thesis in pointing out the chapter/ sections where the content related to the publications is presented or discussed. Further, it shows which ROs are targeted by the presented publications. It can be observed that all the ROs that are specified in Section 1.3 are covered by the respective contributions of my Ph.D. and

presented in order to form a coherent theme for this dissertation.

1.6 Thesis Structure

This thesis is structured into eight chapters, including the introduction chapter. The organization of the chapters and their highlighted content are illustrated in Fig. 1.5 along with the contributions and main contributions. Chapter 2 presents the background knowledge on the MEC technology (i.e., especially the MEC reference architecture and the virtualization design for the MEC edge platforms) and the concept of security/privacy, which is essential to realize the rest of the thesis. Since the introduction chapter (i.e., Chapter 1) specified the research questions and research objectives in regard to the defined problem statement, Chapter 3 reiterates the declared problem in relation to the current literature. In this chapter, MEC security issues and requirements result from my investigations. The same investigation explored the service migration as one of the threat vectors in the considered MEC deployment illustrated in Fig. 3.1. Further, MEC's privacy aspects were presented following my investigations' findings. The chapter concludes by stating that service migration is a critical issue to focus on in MEC, especially considering security. Chapter 4 states the methodology of this Ph.D. and explains the coherent flow of this thesis in regard to the defined research questions and research objectives. Further, enabling factors of the MEC-based service migrations were presented with their validation experiments to cover RO 2.1. Moreover, the main solution of this thesis, the MEC Service Migration Security Framework (MEC-SMSF), is introduced in this chapter, along with its system design, functions, and the life-cycle of a typical migration process.

The primary technical contributions of this thesis are the MEC Service Migration Authentication Protocol (MEC-SMAP) and the MEC Service Migration Security Management (MEC-SMSM) directives. As the underlying infrastructure to launch these two functions, MEC-SMSF plays a key role in aiding to validate these two proposed principle solutions. Chapter 5 explicates the MEC-SMAP protocol design along with the preliminaries (i.e., security goals,

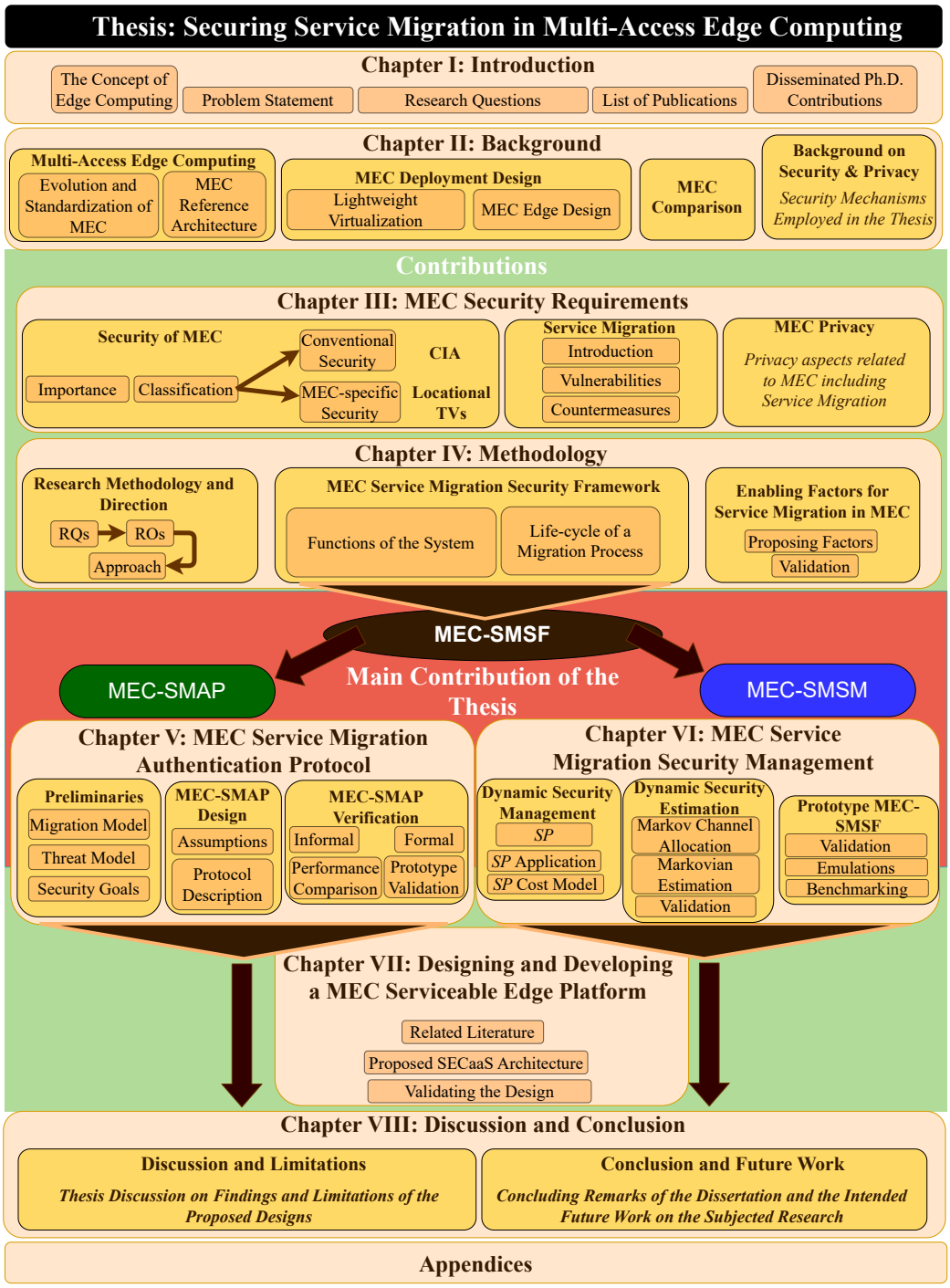


Figure 1.5: Structure and Organization of the Thesis

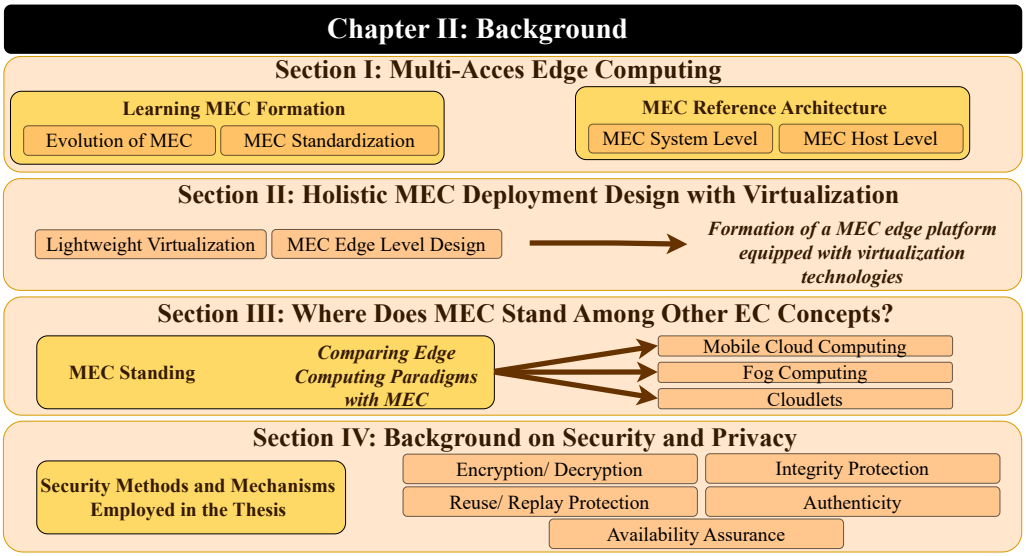
migration model, and threat model) and its verification results. This protocol was verified formally and informally, while performance was compared via simulations, and the feasibility was validated with a prototype implementation. As the other major contribution, Chapter 6 proposes the MEC-SMSM method developed to manage the security in the MEC-SMSF. The theoretical formulation of the dynamic security concept was presented in this chapter, along with the models formed to compute the security cost and estimate the probable security levels as a prediction to enhance the performance of the proposed model. The introduced models were validated through simulations, while the feasibility of the proposed concept was validated with a prototype MEC-SMSF developed for this purpose. Chapter 7 describes the development of a MEC serviceable platform for its edge infrastructure adoption. Security service-based service provisioning architecture was proposed in this chapter while its feasibility was validated in regard to the security service performance. The overall findings of this Ph.D., limitations perceived with the proposed design, conclusions drawn from the work, and the future work are presented in Chapter 8 as the concluding chapter.

CHAPTER

TWO

BACKGROUND

MEC is the preeminent technology targeted by this thesis. As a novel technology, it is promising to elevate the current serviceable infrastructure to the next age by attempting to meet the requirements of emerging applications of 5G and Beyond. It is imperative to understand the background knowledge of the technology before moving on to exposing its security issues. This knowledge is essential to understand the content of the thesis. In this chapter, such background knowledge is presented in the ETSI-standardized context while comparing its features with other edge computing paradigms. Further, literature related to the technologies that can be employed to develop a MEC serviceable platform is presented as the preliminary knowledge for Chapter 7. Understanding security and privacy concepts is vital to read this thesis. Thus, this chapter defines and describes security concepts utilized in this research.



Chapter Organization

This short chapter presents the background knowledge. Section 2.1, describes the technical perspective of MEC as a technology, where its evolution, standardization and its architecture, and pragmatic design possibilities were presented. In this explication, the ETSI-defined MEC reference architecture is presented with its respective functions and purposes. The technologies related to forming a holistic MEC edge computing infrastructure are presented in Section 2.2. Further, the standing of the MEC respective to the other edge computing paradigms is discussed in Section 2.3. Section 2.4 introduced the concepts of security and privacy, while extending the description to the primary security mechanisms used in this research.

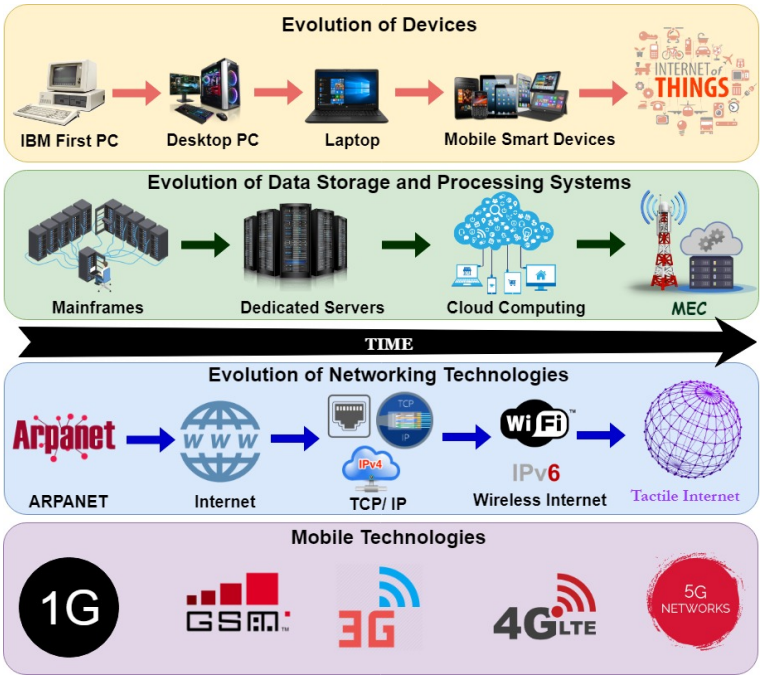


Figure 2.1: Evolution of MEC

2.1 Multi-Access Edge Computing

2.1.1 Evolution and Standardization of MEC

2.1.1.1 Evolution of MEC

The scope of possibilities has been expanded with the invention of Integrated Circuits (ICs) as a paradigm shift that produced the third generation of Personal Computers (PCs) in the 1980s, which revolutionize industries from mechanical processing to electronic-based processing systems. After almost four decades, 5 nm silicon chips are produced to evolve the computers into miniaturized smart devices that perpetrate a higher processing power than attributed to early-era PCs. Thus, the evolution of the computers, data storage/processing systems, and networking technologies depicted in Fig. 2.1 was only feasible with the technological achievements of ICs. Means of computational complex processing are drastically reduced to handheld devices currently, and possibly way diminutive in the future. Thus, managing the data

storage and networking resources in addition to the battery lifetime, computational power, and memory limitations on the end device perspective are disconcerting the Mobile Network Operators (MNOs) [8]. The data storage and processing services have evolved from mainframes (BITNET-1981) to dedicated servers, and to cloud computing while networking services advanced from Advance Research Project Agency Network (ARPANET-1961) to the Internet (1969), to Ethernet, to SATNET, to IPv4-based TCP/IP, to IPv6, to 802.11 Wi-Fi, to 802.11a, to 802.11g, and to 802.11n. Moreover, mobile networks evolved from 1G to 2G - Global System for Mobile Communication (GSM), to 3G, to 3.5G – High-Speed Packet Access, to 4G and to 4G Long Term Evolution (LTE).

The emergence of the IoT paradigm envisaging the any-time anywhere connectivity for myriads of versatile and comparatively miniaturized smart devices was first mentioned by Kevin Ashton in 1999 at MIT; while Chana Schoenberger and Bruce Upbin published the paper titled 'The Internet of Things in Forbes 2002 issue documenting the concept for the first time [35],[36]. However, the drawbacks and limitations in cloud computing exacerbated by the 3.9 billion current internet users are presenting a significant scarcity of capacity from the MNO perspective. Even though the capacity of the core network and cloud infrastructure are upgraded, the existing mobile network limits the accessibility of inflating a number of IoT devices and the launching of impending applications. Thus, the novel approach of MEC is proposed as a paradigm shift to the processing and storage solutions amalgamated with the 5G mobile technology. The networking infrastructure for IoT integrated with the MEC and other edge computing paradigms are envisioned by the Tactile internet concept that guarantees ultra-low latency, extreme availability, reliability, and security provisioned from the 5G and beyond 5G technologies [7].

2.1.1.2 Standardization of MEC

Initially, IBM and Nokia Siemens network introduced MEC in 2013 as a platform that could execute applications within a mobile base station, where

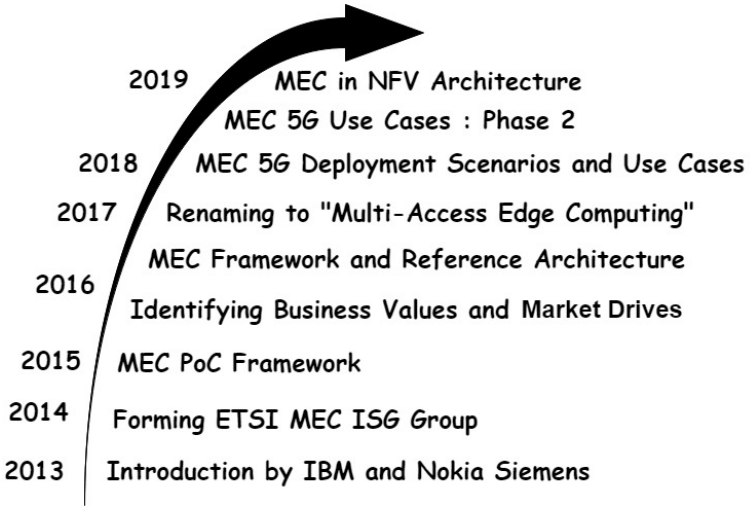


Figure 2.2: Standardization timeline of MEC

the aspects of application migration and interoperability were not considered [37]. Later in 2014, European Telecommunications Standards Institute (ETSI) launched the Industry Specification Group (ISG) for standardizing Mobile Edge Computing (MEC) concept, which specified the operation of MEC as “Mobile edge computing provides an IT service environment and cloud computing capabilities at the edge of the mobile network, within the RAN and in close proximity to mobile subscribers” [38]. A concise timeline of the evolved standards on MEC is depicted in Fig. 2.2.

The pioneers in mobile network solution providers such as Nokia Networks, Intel, Vodafone, IBM, Huawei, and NTT DOCOMO were leading the ISG representation while European 5G Infrastructure Public Private Partnership (5G-PPP) acknowledged MEC as a prime emerging technology for 5G networks [18]. The goal of ETSI MEC ISG was to facilitate an open environment to multiple vendors for provisioning diverse applications and services at the edge of RAN merely to overcome the limitations of existing centralized cloud computing deployments [8]. A Proof of Concept (PoC) framework for MEC was published by ETSI ISG in 2015 for highlighting the rationale, roles, responsibilities, and activation process of the PoC framework [39]. In the same year, another white paper was released for evaluating the business value of MEC service scenarios such as AR, intelligent video acceleration, connected cars,

and IoT gateways to identify the market drives [40]. The MEC framework and reference architecture was published in 2016 by ISG for formulating the entities in the MEC system and to define their intended function [41]. Realizing the potential of the MEC paradigm is reaching beyond mobile networks to Wi-Fi and fixed access technologies, the ETSI ISG group has renamed the concept as “Multi-Access Edge Computing" that conveniently reinstates the acronym to MEC [7]. Moreover, a 2018 ISG release has accentuated the deployment scenarios and use case exemplifications of MEC with 5G integration [42]. This white paper investigates the deployment of MEC and 5G components in actual use cases with traffic steering, mobility, offloading, charging, and regulatory requirement considerations. An extension of the same directive was presented in [43] as the 2nd specification release that expands the scope of MEC use cases for Camera as a Service, video delivery, future factories, multi Radio Access Technology (multi-RAT), Internet Protocol Television (IPTV), In-vehicle systems and 5G use cases. In the latest release from ETSI in 2019 January [44], a variant architecture was released for conjugating Network Function Virtualization (NFV) and MEC while the in-depth focus was drawn towards components in the architecture in contrast to the previous release of [41].

2.1.2 MEC Reference Architecture

The MEC Reference architecture illustrated in Fig. 2.3 is the ETSI-published MEC framework and the reference architecture depicted in [41]. MEC architecture has two main levels: Mobile Edge System Level and Mobile Edge Host Level [25].

2.1.2.1 MEC System Level

User Equipment (UE) is the device that connects to the MEC edge through the access network. They could be either handheld mobile devices or automated devices that are operating without human supervision. UE Applications (UE Apps) are programs intended to be subscribed by MEC services and executed in single or several UE domains. The connectivity of UE Apps to the

MEC host level, however, is commandeered by the User Application Life-Cycle Management Proxy (UALCMP). UALCMP handles the initial UE App requests for subscriptions. Though UALCMP is located at the mobile edge system level, subscription requests conveyed through the access network are forwarded to it from the edge network. A similar function is performed by the Customer Facing Service (CFS) Portal for third-party customers of the MEC service provider or the mobile operator for facilitating the MEC services. The Operation Support System (OSS) is the entity that handles the user access authorization and subscription elapsing duration distinguishing the service types forwarded from UALCMP and CFS portals. Additionally, connections are maintained with the mobile edge platform manager and the mobile edge orchestrator for virtual resource allocation to subscribed UE Apps and provisioning service logs, respectively. The Mobile Edge Orchestrator (MEO) is the principal entity at a mobile edge system level that governs single or several mobile edge host levels. The MEO is linked to the OSS and UALCMP entities operating at the MEC system level, while connections are extended to the mobile edge platform manager and virtualization infrastructure manager at the MEC host level. The holistic management of operating MEC hosts, catered services with resource utilization, and employability of the standardized topologies, are administered by the MEO.

2.1.2.2 MEC Host Level

Mobile Edge Platform Manager (MEPM) acts as the orchestrator for the mobile edge host level. MEPM manages the rules, requirements, and life-cycle of mobile edge applications by handling the storing, configuring, and running functions of software images in the host virtual environments. The virtualization Infrastructure Manager (VIM) governs the virtualized resources in every mobile edge host launched at the mobile edge host level. It is connected to the virtualization infrastructure of each mobile edge host, while the status of the virtual resources is updated to the MEO and MEPM. Mobile Edge Host (MEH) is the executing entity of MEC services, whereas all other entities are designed to mandate various monitoring and approving functions that ensure

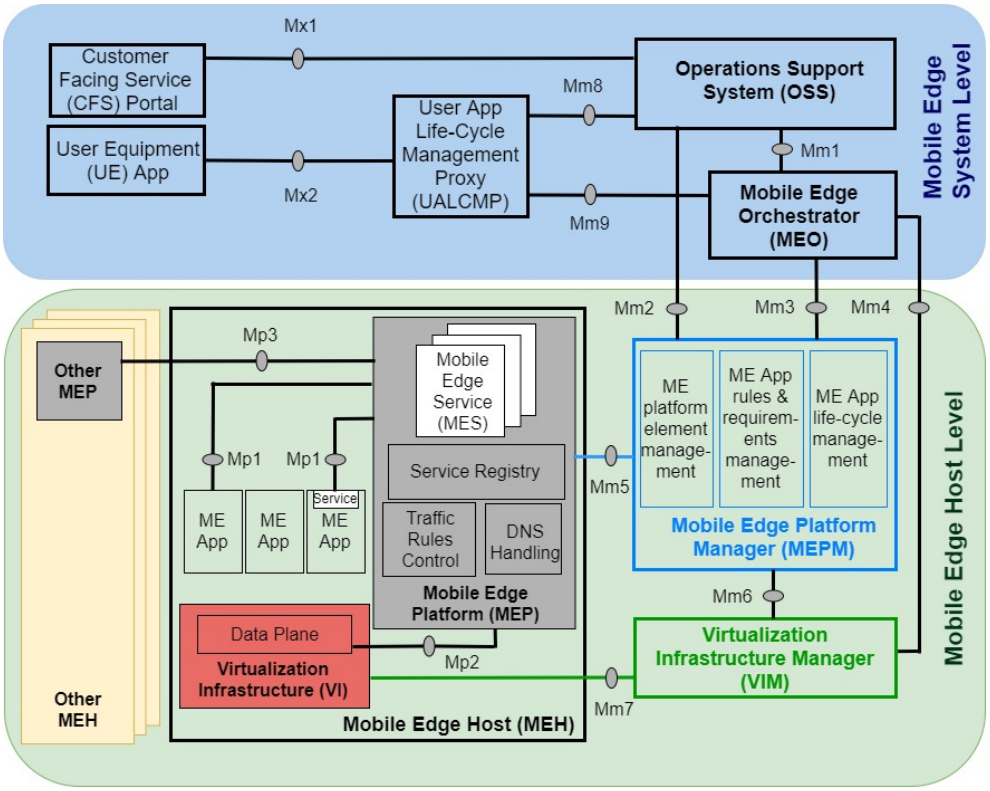


Figure 2.3: MEC reference architecture

the seamless operation of the holistic MEC system. A MEH contains MEC applications, a mobile edge platform, and a virtualization infrastructure. Mobile Edge Applications (ME Apps) are software-based processes that operate as Virtual Machines (VMs) on top of the virtualization infrastructure. The nature of the ME App in terms of its VM configuration (storage, processor, and networking) and connectivity (to other ME Apps, to ME Apps in other MEHs, or to another mobile edge host level) is reliant on the scope of the subscribing UE App and intended MEC service. ME App connectivity is established from a Local Area Data Network (LADN) that extends within a single MEH [42]. The Mobile Edge Platform (MEP) dispenses the provisions to launch the ME Apps while creating an environment to discover, advertise and consume mobile edge services. The MEPM-condoned traffic rules are notified to the LADN at the MEH by the traffic rules controller, where DNS and proxy functions are administered based on the MEPM records. Virtualization Infrastructure (VI) is

the platform on which the ME App VMs are functioning. The VI comprising the LADN is commandeered by the VIM and MEP for managing virtual resources and enforcing traffic rules, respectively. In relation to the MEC system, a Mobile Edge Service (MES) is defined as a service originated or facilitated by either the MEP or ME Apps. All the provisioned services are registered under the MEP through the Mp1 interface, where the ME App is subscribing to the relevant authorized services. More details regarding the interfaces in Fig. 2.3 are explicated in [41].

2.2 Holistic MEC Deployment Design with Virtualization

Virtualization is an imminent technology for launching MEC in a highly efficient and autonomous environment with zero human intervention. In order to guarantee ultra-low latency and ultra-reliable services, automation is the way forward. Virtualization is not a novel concept, and current cloud-based service platforms are depending on virtualization technologies to maintain their Platform as a Service (PaaS) services. Though, virtualization technologies are evolving rapidly to cater to dynamic and low-overhead service instances that are more flexible and task-specific than common Virtual Machines (VMs) launched through hypervisor-based virtualization. VMware, VirtualBox, and Hyper-V are typical hypervisor-based virtualization tools that can create VMs either as a Guest Operating System (OS) on top of the host OS or without an OS, where the condition is termed bare-metal virtualization.

2.2.1 Lightweight Virtualization

In contrast to hypervisor-based virtualization, lightweight or containerized virtualization operates with a container engine on top of the host Operating System (OS). The containers are executing independently of an OS kernel compared to guest OSs on a hypervisor. This is the salient feature that enables dynamic loading, attachment, and detachment of container images on the con-

tainerized infrastructure. These self-contained, properly isolated, and highly-portable virtual entities are facilitating cost-intuitive users with a wide range of applications, bins, and libraries adaptable to their needs. In this thesis, Docker virtualization is considered as a lightweight solution due to its wide adaptability in cloud environments.

Docker is a prominent lightweight virtualization technique that is capable of launching at computing platforms with alleviated resources. Its ability to execute multiple service instances with minimal resources enables its integration to emerge virtualization-based systems. Docker is formed as a client-server model, where the docker daemon is handling the user requests interfaced by a docker client. The extracted images from the docker registry are executed as containers within the docker host. Each container is given an ID and is capable of committing the status of the container as a newly forged image. These containers are connected with a default bridge network. The overhead is remarkably lesser than 70 MB for a Ubuntu-based container. Thus, deploying multiple containers at an edge device is no longer an arduous task. Moreover, [45] presents that Docker containers are fairly secure in their default configuration while the level of security could be enhanced with hardening solutions accessible through Linux kernels. In the Dockerized environment, docker containers are executed at the docker host with the images loaded from the registry. Users communicate with the docker daemon via the docker client.

2.2.2 MEC Edge Level Design with Virtualization Technologies

Fig. 2.4 illustrates the design for a MEC system that adheres to the ETSI-defined MEC architecture presented in Fig. 2.3. These design options were explored in my research published at [15]. According to the design, system-level entities can be hosted at cloud-native or stand-alone data centers belonging to the MNO. MEC edge level can be launched in a bare-metal hypervisor-enabled (VMware ESXi) high-performance server, that can be installed at a MEC-enabled gNB with the interfacing capability to the RF apparatus of the gNB. The MEPM functional instance can leverage the orchestrator options

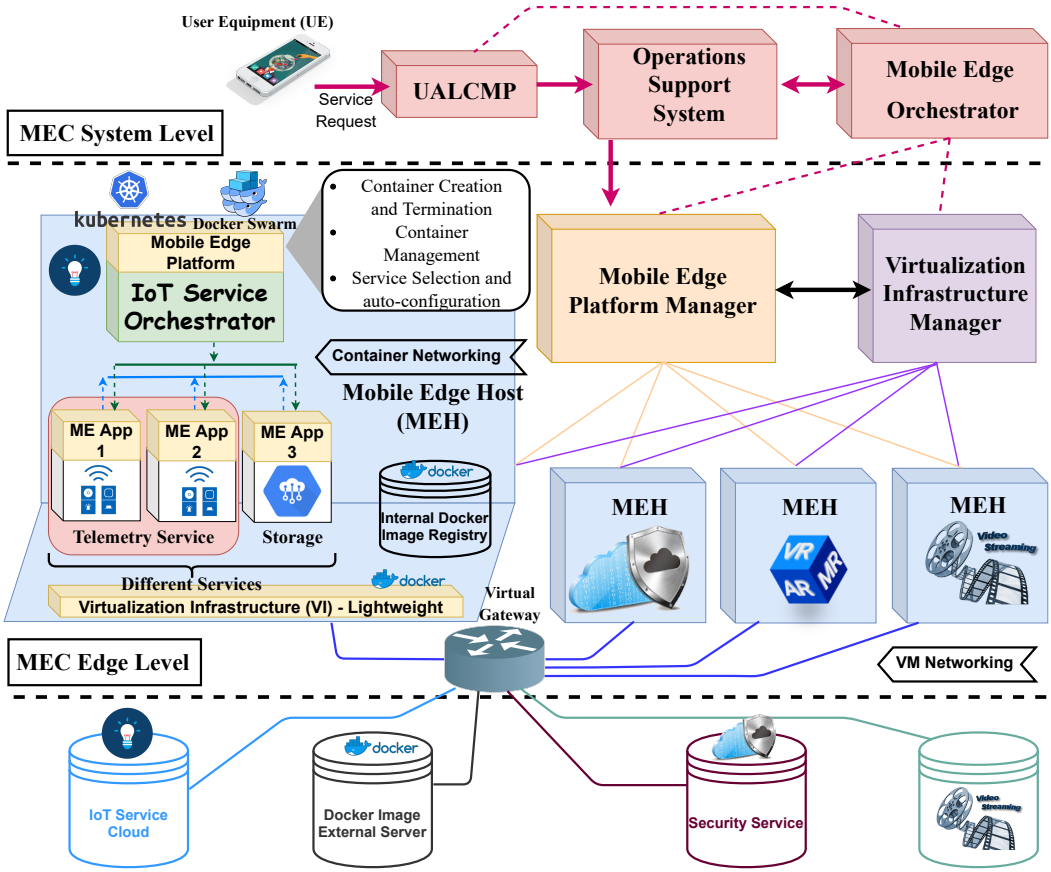


Figure 2.4: Technical Perspective on MEC System Design with Virtualization Technologies

available with the launched hypervisor technology. All the hypervisors have their own in-built virtual resource monitoring function, that can be leveraged for VIM. If each MEH is assumed to be representing a distinct MES, then each MEH can be launched as a VM hosted by the deployed hypervisor. Within the MEH or the VM, a Lightweight Virtualization Infrastructure (LVI) can be contrived by leveraging Docker technology. In this context, either Docker Swarm or Kubernetes technologies can represent the MEP function, where container creation, monitoring, management, and termination can be performed through them. The ME Apps can be launched as docker containers that are specifically configured to host the specified service function. As in Fig. 2.4, telemetry and storage service related to an IoT service is hosted at the MEH, where its exter-

nal IoT service cloud is connected to the MEH through the virtual gateway of the MEC-edge. Similarly, other services can host different MEHs for launching their service instances at this subscriber model. This design was followed in the publication [32], where a prototype MEC edge platform was constructed and demonstrated for its autonomous subscriber request handling for MEC services. Further, the approaches presented in [46], and their management of service downtime within the 3.5 seconds window for service migrations employing ETSI MEC Sandbox improve my designs' feasibility.

2.3 Where Does MEC Stand Among Other EC Concepts?

Tables 2.1 and 2.2 present the various features/ attributes of the MEC technology and their contrast against the other existing EC technologies of MCC, Fog, and cloudlets. Accordingly, MEC and Fog computing are leading in terms of the feasibility of deployment.

2.4 Background on Security and Privacy

Security is a broader concept that extends to the notions of information security, cyber-security, forensic security, and network security. Information security was defined as the preservation of confidentiality, integrity, and availability (also referred to as the CIA triad) of the information under the standard ISO/IEC 27002 in 2005 [47]. The information under this definition is applicable to physical or electronic/digital forms of data that are subject to be documented, stored, in transit, or processed. Forensic security covers acts committed against the laws and statutes in the governing domain. In the IT domain, however, digital forensic methods are used to ensure security.

Table 2.1: Comparison of edge computing paradigms 1

Factor / Technology	**Multi-Access Edge Computing (MEC)**	**Mobile Cloud Computing (MCC)**	**Fog Computing**	**Cloudlets**
Introduced by	ETSI (2014) [38]	Aepona (2010) [7]	Cisco (2012) [48]	Satyanarayanan et al. (2009) [49]
Standardized by	ETSI, 3GPP, ITU-T [50]	NIST [50]	OpenFog Consortium, IEEE [50]	OpenEdge [50]
Purpose	Extending cloud computing capabilities to the edge network [7]			
Infrastructure Ownership	Telecom MNOs [7]	Private Institutions and Individuals [7]		Private Institutions [37]
Node Deployment	At the RNC or BS [7]	Network edge [37]	between cloud and device stratums [7]	Network core [37]
Software architecture	MEO based [7]	Service Oriented [7]	Fog abstraction layer based [7]	Cloudlet agent based [7]
Virtualization	VMs or other virtualization techniques [48]	Only VMs [51]	Other virtualization technologies [48]	Only VMs [48]
Operation Mode	Standalone or Cloud Connected [48]	Cannot work Standalone [51]	Cannot work Standalone [48]	Only Standalone [48]
UE Access	Closest RNC or Access Point [7]	Internet [7]	Closest RNC or Access Point [7]	Closest Access Point [7]
Latency and Jitter	Very Low [7][50]	Relatively High [8][50]	Very Low [48][50]	Very Low [37][50]

Table 2.2: Comparison of edge computing paradigms 2

Factor / Technology	**Multi-Access Edge Computing (MEC)**	**Mobile Cloud Computing (MCC)**	**Fog Computing**	**Cloudlets**
IoT Compatibility/ Adaptability	High [14]	Low [8]	High [48]	High [7]
Storage Capacity at the edge	High [14]	High [7]	Depends on the deployment [52]	
Power Consumption	High [50]	Low [50]	Low [50]	Moderate [50]
Computation power at the edge	High [14]	High [7]	Depends on the deployment [52]	
Availability	High [37]			
Scalability	High [37]			Low [37]
Mobility	High [37]	High [37]	High [37]	Low [37]
Context Awareness	High [7]	High [7]	Medium [7]	Low [7]
Local Awareness	High [37]			
Security	High	Medium	High	Medium
Integrating Technologies				
NFV	✓[53]	✓[54][55]	✓[56]	✓[57]
SDN	✓[53]	✓[54]	✓[58]	✓[59]
Network Slicing	✓[53]		✓[60]	
ICN	✓[53, 61]	✓[55]	✓[62]	✓[63]

A more nascent definition for cyber-security is presented in [64] as the approaches and actions associated with security management processes followed by organizations and states for protecting CIA data and assets in cyberspace although the latest requirements of cyber-security are going beyond CIA aspects. In fact, the protocols conducting the authentication and the access control systems governing the authorization aspect of a system have to incorporate measures exceeding CIA defense mechanisms to overcome novel security threats of masquerading, virtualization-based, and service denying. In addition, traceability, anonymization, granularity, localization, and trust are novel requirements for systems where cyber-security is applicable.

Initially, network security was defined as the means to secure communication networks from possible intrusions and vulnerabilities. Those attacks and threats were limited to intervening and masquerading attacks such as Man-in-the-Middle (MitM), Relay, and spoofing. With adequate levels of encryption and cryptography primitives, probable attacks were plausibly mitigated. However, novel communication services are prioritizing the data rate of the network to serve more subscribers. Thus, cumbersome cryptographic primitives are undesirable. Moreover, softwarized approaches of Software Defined Networking (SDN), Network Function Virtualization (NFV), and Network Slicing demand more requirements for security assurance as presented in [65]. Most of the emerging systems are Cyber-Physical Systems (CPS) that integrate computation, networking, and physical processes to create an environment extending to cyber and physical spaces. Thus, security for a CPS represents an extensive domain for cyber, information, forensic, and network security contexts.

Privacy is an individual's right to act or behave independently of any records or surveillance activity conducted without their consent. In the digital context, personal data cannot be mishandled by service providers without their consent, and measures should be taken to keep safe a user's identity, while the user's actions should be untraceable. Irresponsible entities possessing the personal data of their consumers might opt to outsource them to an external institution for deriving personal intents, behaviors, or interests to expand their commercial market. Furthermore, adversaries are capable of extracting per-

sonal credentials from weakly protected systems to violate their privacy. These acts are recognized as unlawful practices, and novel legislations are focused on mitigating these occurrences. Advancement of sensory devices appended to both human and non-human entities is increasing the possibility of privacy leakages [3].

2.4.1 Security Methods or Mechanisms Employed in the Thesis

Security cannot be achieved by a single action and require multiple functions conducted to maximize the protection of the secured content. In other words, multiple security mechanisms are required to satisfy multiple security requirements. The following sections list the employed basic security measures of this thesis. Extended technical information about these security mechanisms can be found under Section 5.3.3.

2.4.1.1 Encryption/ Decryption

Encryption is defined as obscuring the plaintext by cryptographic means to invoke confusion and diffusion. This is the primary means of applying security. In general, the plaintext is processed through a cryptographic encryption algorithm to produce an outcome called ciphertext. This ciphertext is then conveyed or stored instead of plaintext. And the same ciphertext is used to recover the plaintext using the same cryptographic algorithm but using its decryption mode. An important parameter called a key should be inputted into the algorithm, which has to be a secret for this process to succeed. The encryption algorithms are typically categorized into symmetric or asymmetric methods, depending on whether the same key or different keys are used in encryption and decryption processes. This thesis uses the RSA, Advance Encryption Standard (AES), and Elliptic Curve Cryptographic (ECC) algorithms/ methods for the security protocol-related designed solutions.

2.4.1.2 Integrity Protection

Since integrity protection involves detecting and preventing any unauthorized modification or destruction perpetrated on the securing asset, it is as vital as encryption. Both symmetric cryptographic algorithms or hashing functions are employed for the detection of integrity violations by attaching a cryptographic residue of the plaintext called a message authentication code to the conveying message. Those residues are checked at the receiving end for detecting violations.

2.4.1.3 Reuse/ Replay Protection

These are threats where an attacker attempts to reuse a captured message by resending it at another time. Timestamps are the popular parameters that are used for detecting any reuse or replay attempts. For timestamp-based implementations to succeed, the subjected systems should be synchronized to avoid time lapses.

2.4.1.4 Authenticity

Identity assurance or authenticity of the entities engaged in communication is a primary function of any security system. In fact, it is called authentication. Authentication is typically achieved through well-defined protocols, where each party validates their identity, either through a trusted third party or other means. Various practices are followed to ensure authenticity. Typically, asymmetric encryption, nonces, timestamps, perfect forward secrecy, and signatures are integrated into a protocol to verify the identities of the parties engaged.

2.4.1.5 Availability Assurance

Denial of Service-based attacks are the common form that can compromise the availability of a system. The detection of such attacks leads to the prevention of these attacks. This thesis employs a puzzle-based challenge-response mechanism for detecting DoS attempts [66].

CHAPTER

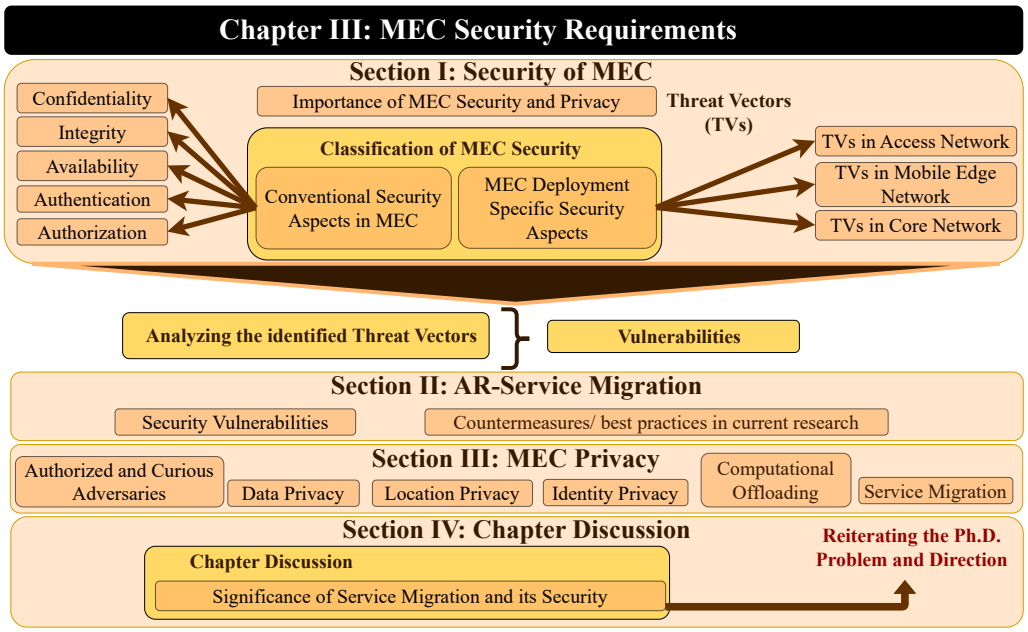
THREE

MEC SECURITY REQUIREMENTS

In this chapter, a thorough investigation of the MEC architecture in a deployment scenario is conducted while the identified threat vectors are analyzed extensively. The vulnerabilities leading to the identified threat vectors are analyzed and potential security solutions are proposed to overcome these vulnerabilities. Moreover, a discussion is presented on the critical security issues of MEC, where service migration has been emphasized as the directive for this study with driving research questions. This review was essential in the context of this thesis for identifying the Ph.D. problem and defining the respective RQs and ROs.

Chapter Organization

This chapter is divided into four sections. In addition, a review of the current virtualization technologies is presented to facilitate the possible deployment perspective to the reader. In Section 3.1.1, the conventional security aspects of MEC are presented. Section 3.1.2 introduces the TVs specific to MEC systems based on a deployment scenario illustrated in Fig. 3.1. Each stated TV is analyzed for exposing its vulnerabilities.



Section 3.2 introduces the concept of service migration, exposes its security issues in the context of the MEC deployment, and proposes the countermeasures from the literature to mitigate those flows. A discussion on the privacy of MEC is presented under Section 3.3, where several aspects, including service migration, are listed. Section 3.4 summarizes the assimilated knowledge and discusses the significance of service migration along with its security in the context of MEC. This discussion leads to reiterating the declared problem statement of this dissertation.

3.1 Security of MEC

The edge of the mobile network is the access point to all the services emanating from the RAN. This critical juncture is one of the weakest points in the entire network in terms of security. The majority of Internet of Things (IoT) devices in the market are produced with economically manufactured circuitry that employs weak encryption/encoding schemes and other security measures to maintain an affordable price range in order to compete. IoT devices are vul-

nerable to cloning and physical tampering, imperiling the entire mobile network for countless attacks. Tamper-poof mechanisms, however, are introduced to the IoT devices that are available in the market [67, 68]. Though, the majority of IoT devices do not have the resources to employ such tamper resistance mechanisms, especially a technology such as Blockchain. Thus, verifying the credibility of these devices at the edge is a major concern. In addition, the distributed nature of the MEC paradigm is broadening the avenues for adversaries due to the migration of storage and processing service infrastructure to a proximate radio access range. Even a service impeding attempt threatens the purpose of MEC, for attaining ultra-low latency to provision real-time 5G-based services.

Impending applications and services are demanding the handling of personal credentials/information at the edge of the network for realizing the service requisites. Privacy, integrity, and trust management assurances are prime requirements with MEC deployments, despite the attributed locational and contextual awareness facilitated for the users. It is evident that virtualization technologies are vital for realizing the MEC paradigm and for creating a serviceable platform with dynamic resource allocation capability. Security of the virtualized platforms is still a gray area due to lesser deployments. The vulnerabilities and attacks plausible on Virtual Machines (VMs) are unique and cause significant consequences to the MEC system.

Similar to CC, outsourcing MEC subscriber data to a remote storage and processing environment creates a predicament in terms of privacy rights. Establishing boundaries regarding the extent of authorized conduct on service providers' capabilities is imperative for guaranteeing the trust of MEC consumers. Considering all these facts, security and privacy are important for realizing a pragmatic MEC paradigm deployment.

3.1.1 Conventional Security Aspects in MEC

This sub-section states the classical security aspects of MEC deployments. In addition to the CIA triad, authentication and authorization have been included as separate aspects due to the limitation of conventional CIA concepts to ad-

dress all the required security requirements in novel applications, as stated in Section 3.1. These aspects can be conceptualized as requirements to improve the feasibility of launching MEC.

3.1.1.1 Confidentiality

Confidentiality is the act of preventing unauthorized entities from reading or accessing sensitive materials [33]. This aspect of security obscures information by encrypting the payload with a considerable level of cryptography. At the design stages, obscuring information that ingresses, egresses, and traverses within the MEC system are a critical requirement. This will prevent the disclosure of information by intervening and eavesdropping attacks. Mobile networks evolving from GSM to LTE employ encryption algorithms ranging from A5/2 to Evolved Packet System (EPS) Encryption Algorithm (EEA) [69]. Moreover, security specifications published under TS 23.122 and TS 33.210 guarantee the security measures available at Access Stratum (AS) and Non-Access Stratum (NAS) levels of 3GPP architecture [4]. The insights gained from the prevailing network models enhance the security of mobile protocols in regard to confidentiality.

3.1.1.1.1 Possible Confidentiality Violations in MEC For 5G and beyond-5G-based RAN, however, communication protocols should be customized in accordance with the deploying use cases and applications. The information traversing within the edge infrastructure and towards the core network should be encrypted with the security level of TOP SECRET [70] (i.e., AES-256). Unwarranted information disclosure at this level imposes more damage to the MEC system than the mobile AN via exposed system states and cryptographic primitives. Non-3GPP-based IoT devices face the same threat levels as mobile UEs. However, their threat domain exceeds a typical UE device due to their resource scarcity and inability to launch adequate security measures.

3.1.1.1.2 Mitigating Confidentiality Violations Both signaling and controlling information related to the mobile network and virtualization platform are conveyed through the link between the edge and core levels as described under TV E5. For such links established between edge infrastructures; tunneling, IPSec, or TLS/SSL-based encryption schemes are viable adaptations for guaranteeing End-to-End (E2E) security [18]. Slice isolation is a way of ensuring the confidentiality of UEs sharing the same infrastructure, accessed through the RAN [71]. For Non-3GPP devices employed in Zigbee, Bluetooth, Wi-Fi, and LPWAN technologies, mechanisms such as lightweight security protocols, ECC, and PHY schemes are emerging methods for improving security[72].

In addition, Quantum Resistance (QR) or anti-quantum cryptographic methods are being researched to limit the advantage of adversaries with ample resources [73]. Lattice-based, multivariate, hash-based, and elliptic curve schemes are employed for formulating QR algorithms in order to overcome exhaustive key searching or brute-force attacks commandeered by quantum computers [23]. These methods guarantee integrity in addition to confidentiality.

3.1.1.2 Integrity

Manipulation and destruction of data to mislead the parties engaged in communication are integrity violations. Similar to confidentiality, integrity is a widely addressed concept for mobile networks. The virtualized MEC edge platform relies on control information conveyed through feedback channels to optimize the operations of virtual entities. Therefore, integrity plays a key role in the MEC context as these services are automated, and autonomous services require accurate information to operate effectively.

3.1.1.2.1 Possible Integrity Violations In mobile networks, integrity is violated through intervening attacks such as MitM or Relay attempts perpetrated to manipulate or misuse the inwardly and outwardly conveyed content to and from the core network. In MEC, additional channels are being exposed to the adversary in contrast to a typical mobile network. The links established

between the MEC edge level, MEC system level, and the mobile/5G core network adds more threat vectors for violating integrity. The injection of malicious codes into a legitimate information flow can cause the most devastation to the edge platform [74]. Reliance on the softwarized core and edge platform entities for autonomous operation risks compromising the entire system through such a malicious fragment. All operations conducted by an illegitimate node, or a device inserted into the network, either in the access network or within the edge platform, can be considered integrity violations.

3.1.1.2.2 Mitigating Integrity Violations Through EPS Integrity Algorithm (EIA), integrity protection is offered to AS and NAS stratum levels in LTE [4]. In 5G, 5G-AKA is handling the integrity assurance mechanisms for signaling channels [75]. Moreover, tenant isolation is a prospect that improves integrity protection in a multi-sliced environment [71]. In comparison to LTE, 5G is supporting integrity at the user plane [4]. This feature enables resource-constrained IoT nodes for utilizing integrity verification mechanisms available under 5G RAN. For IoT devices, hash-based and session-key-based encryption schemes could be employed for ensuring integrity at the PHY level [76].

Typically, edge and core levels are linked with communication channels that use existing protocols. The Encapsulation Security Payload (ESP) attribute of the IPSec is used for the integrity protection of IPSec-based tunnels which are adaptable for egressing channels of MEC edge[77]. In addition to traversing information, the integrity of applications or MESs should be validated routinely. An Operation Support System (OSS) is capable of initial integrity verification, while MEO can monitor the integrity in softwarized entities deployed at the edge infrastructure [78].

3.1.1.3 Availability

Availability is the omnipresence of MEC resources for consumers who are willing to subscribe to services. This factor primarily relies on network performance and the effectiveness of the network interfaces. Therefore, the performance of the mobile network is paramount for MEC.

3.1.1.3.1 Possible Availability Violations DoS-adjacent attacks that impede services are the main cause of availability disruptions in communication channels. In the existing LTE network model, the Radio Resource Control (RRC) connection status creates issues with validating the eNodeB for the UE [79]. Thus, it creates opportunities for DoS attacks, compromising the availability of eNodeBs. Since the UE requests are directed to the UALCMP initially, its capacity for serving UE should improve significantly to cater to the 5G-based applications. Once the service request is granted, UE establishes a direct link with the MEC edge for instigating its service. This channel is formed via the RAN and connects to the MEH via the UPF instance in the LADN. Thus, the N3 interface of the 5G service architecture should feature the adequate capacity to handle minimum UE requirements [42]. Failing to integrate these requirements into the novel networking interfaces compromises the network flow and results in inaccessibility. Further, the nature of service denial attacks has evolved drastically over the years, delivering distributed attacks with a multitude of bots. These bot-net-type attacks limit the accessibility of legitimate users, compromising their availability.

3.1.1.3.2 Mitigating Availability Violations In the edge infrastructure, the placement policy of virtualized entities plays a key role in maintaining availability parameters. However, MEPM and VIM are static placements. Depending on the service provided by VMs as isolated hosts or VMs deployed within a singular host, each represents different availability and cost factors [80]. As the virtual environment formed by the ME Apps within a MEH can be customized according to the service type, various ME Apps should function distinctly from each other. Therefore, the placement of ME Apps within the MEH directly affects the availability factor.

3.1.1.4 Authentication

Authentication is the process of verifying the identity of the parties engaged in communication or resource access. These mechanisms are either performed by a single party or mutually through an extended scenario. Authentication

schemes employ various measures of authenticity for validating the entities. Keys are the generic tool employed for authenticating non-human entities. Depending on the domain where the authentication is instigated, the mechanisms are classified as either primary or secondary authentication. The authentication for MEC-based UEs is handled via the air interface of mobile RAN mostly as a primary approach. Thus, heterogeneous IoT devices and services incur diverse authentication requirements. Ensuring the confidentiality of keys and authentication credentials is intrinsic for UEs, core-level, and edge-level entities. UE protection can be acquired through the enhancement of existing EPS Authentication and Key Agreement (AKA) mechanisms employed by LTE [79]. Further, 5G-based AKA and Extensible Authentication Protocol (EAP) AKA are two mandatory authentication schemes proposed under 5GPP phase 1 [4].

3.1.1.4.1 Possible Authentication Violations There are various ways that the authentication phase of a MEC-based service can be compromised. In primary authentication, device cloning and masquerading attacks of spoofing and impersonation are viable and common through the air interface. Further, Evil Twins (ETs) and injection attacks are plausible, in addition to the previously mentioned attacks in Device-to-Device (D2D) scenarios - where a compromised node can be authenticated as a legitimate entity to the MEC system [81]. The autonomous and virtualized edge platform is operating with virtualized entities (MEHs) that require continuous authentication with UEs, MEHs in other edge platforms, or external cloud services. Thus, adversaries can target these authentication sequences to gain access to the system. Moreover, UALCMP and CFSP, as the main authentication handling entities in the MEC system, can be subjected to DoS or DDoS-type attacks perpetrated through authentication requests.

3.1.1.4.2 Mitigating Authentication Violations For most IoT devices, non-3GPP-based technologies are employed for communication. Wi-Fi is a common technology used in most edge computing circumstances because of its range. Methods such as PUF [82], accelerometer data [83], and visible light (referred to as Li-Fi) [84] are explored for improving the authentication of

Wi-Fi networks. Moreover, novel methods are introduced for securing LPWAN [72], NB-IoT [85], RFID [86], and BLE [87] authentication phases. These technologies are used for D2D or ad-hoc type authentication, which is common for IoT deployments. QR authentication is a directive that would benefit resource-constrained devices due to its cryptographic primitives bearing a lesser overhead [88]. Thus, such schemes are viable for IoT-based technologies that interface with the MEC system.

3.1.1.5 Authorization

Authorization is the function of granting access to authenticated entities, classified under diverse capability levels. Depending on the service type the UE is requesting, OSS approves the capability level for the specified ME App at the edge level. In addition, MEPM is responsible for restrictions imposed by the OSS for ME Apps. Thus, MEC already possesses an authorization discipline devised within its architecture. However, a proper authorization framework should be identified from the existing deployments to be compatible with prevailing mobile services.

3.1.1.5.1 Possible Authorization Violations Most authorization violations are instigated as authentication violations. OSS, as the main authorization handler for assigning virtual resources to the MESs, can be misled by illicit UEs with granted access. These UEs can get approval to utilize massive amounts of the edge platform's resources, leaving them scarce. Privilege escalation is an obvious repercussion of an authorization violation that applies to malicious UEs and virtual entities operating at the mobile edge [89]. A compromised MEH is capable of overloading the MEC system-level entities of OSS and MEO with security and service log manipulations.

3.1.1.5.2 Mitigating Authorization Violations Mechanisms should be implemented for handling security logs while detecting illegitimate log entries to identify malicious entities [89]. A Trusted Platform Manager (TPM) can be employed to detect illicit entities through their performance metrics. Further,

5G AKA EAP standards include preventive mechanisms for access controlling that furnish an authorization framework for the mobile network [90]. Violations of authorization acts can be mitigated through a secure authentication mechanism. The MEPM, acting as the orchestrator for the edge platform, is responsible for commandeering the access control operations securely [91]. Further, Blockchain can be employed for securing the authorization handling framework of the MEC system to minimize violations while maintaining system logs securely [92].

Table 3.1 classifies the identified solutions in Section 3.1.1 for the security requirements.

3.1.2 MEC Deployment Specific Security Aspects

In this sub-section, Threat Vectors (TVs) existing in the considered MEC deployment scenario derived in compliance with the ETSI standardized architecture presented in Fig 2.3 are presented. Fig 3.1 illustrates the locational TVs applied at different levels in the MEC structure. They are categorized into three areas based on their scope of intrusion: access network, mobile edge network, and core network. In addition, TVs that are not locational, are further classified as architectural and other TVs for understanding their applicability.

3.1.2.1 Threat Vectors related to the Access Network

The Access Network (AN) represents the RAN defined by the 3rd Generation Partnership Project (3GPP) and the access infrastructure related to non-3GPP networks, such as Wireless Local Area Network (WLAN), Worldwide Interoperability for Microwave Access (Wi-Max: IEEE 802.16), or Code Division Multiple Access (CDMA) 2000 networks [93]. In any communication system, threats mostly originate in the access network. The prime rationales for those threats are the diverse technologies deployed in the AN scope. The scalability of the proposed TVs based on AN is a great concern for applicability, due to the heterogeneous nature of the MEC-enabled services. TVs related to AN are classified into three generic categories, described in the following subsections:

Table 3.1: Classification of solutions for conventional security aspects

Aspect	Solutions	References
Confidentiality	E2E IPSec or SSL tunneling	[18]
	Slice isolation	[71]
	ECC, PHY based light-weight security protocols	[72]
	QR cryptography	[73]
Integrity	EIA integrity protection for LTE	[4]
	5G AKA for signalling integrity	[75]
	Tenant isolation in multi-slice environment	[71]
	Hash or session key-based integrity assurance at PHY layer	[76]
	ESP attributed IPSec tunneling	[77]
Availability	Optimum placement of ME Apps within the MEH	[80]
Authentication	PUF	[82]
	Accelerometer data	[83]
	Li-Fi data	[84]
	LPWAN authentication	[72]
	NB-IoT authentication	[85]
	RFID authentication	[86]
	BLE authentication	[87]
	QR authentication	[88]
Authorization	Detecting fake security logs	[89]
	TPM for log validating	[89]
	5G AKA EAP for access control	[90]
	Blockchain for authorization framework	[92]

A1, A2, and A3; and illustrated in Fig. 3.2.

3.1.2.1.1 A1 - Link between the User Equipment and a Base Station The connection from UE to the BS is the most typical communication link that exists and the most vulnerable to threats in a mobile communication system. Since A1 applies to the most exposed part of the mobile communication network, an adversary could either intervene or introduce a malicious device to compromise the BS. Mobile delegation is concerned with offloading computa-

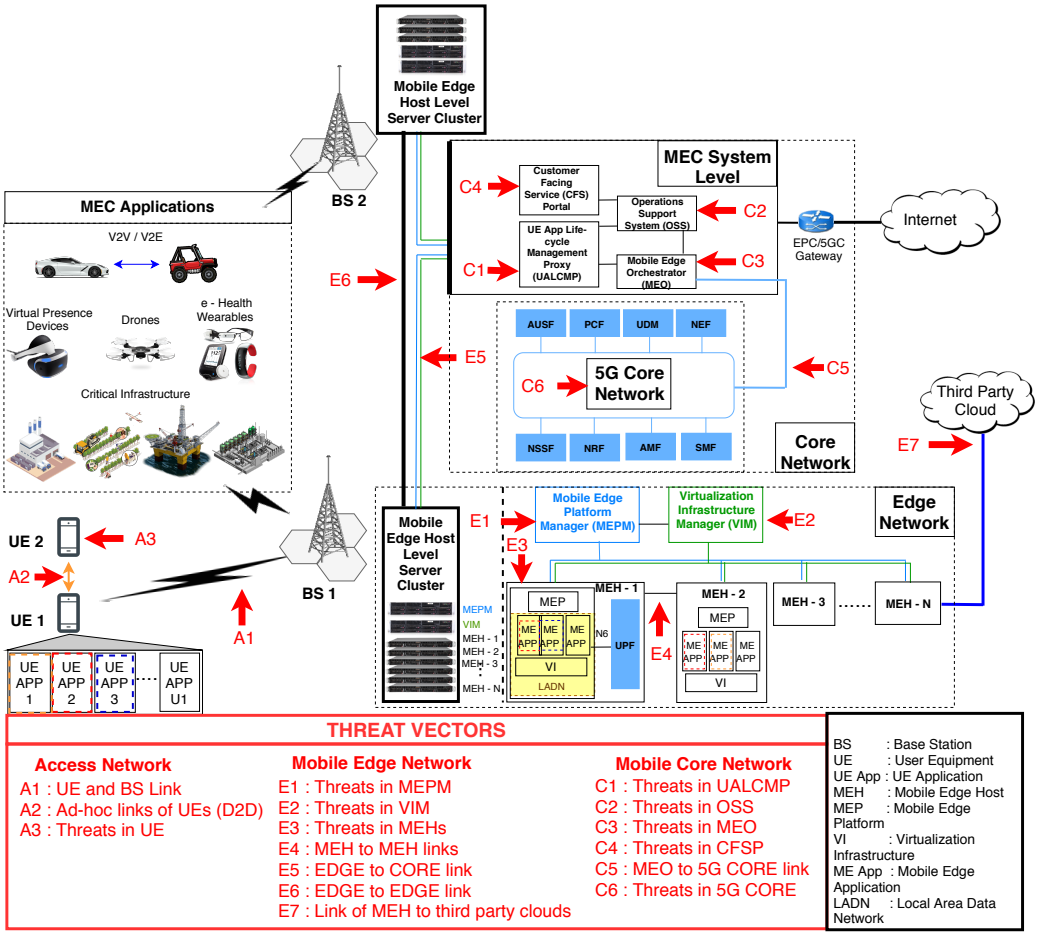


Figure 3.1: Locational threat vectors of a typical MEC deployment.

tionally intensive tasks to the edge servers due to resource constraints of the UEs. The bulk storage and information traversing intended for processing in an offloading scenario are elevating the network traffic carried through the air interface [37]. Additionally, novel technologies such as massive Multiple-Input-Multiple-Output (MIMO), interference-aware receivers, advanced coding/modulations, millimeter Wave (mmWave), carrier aggregation, Wi-Fi offloading, LTE and License Shared Access (LSA) have been introduced in order to improve the spectral efficiency in the access network [94]. The connectivity of the UE through these heterogeneous technologies raises concerns over compatibility and interoperability factors that could be exploitable by adversaries.

Vulnerabilities: The wireless, broadcast nature of the air interface that

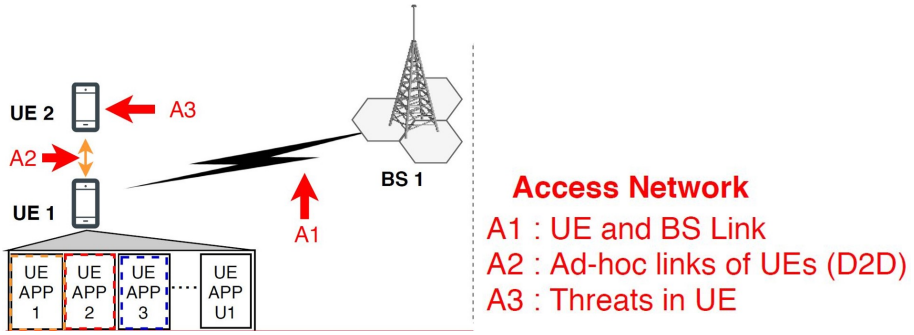


Figure 3.2: Threat Vectors in the MEC Access Network.

links the UE to the BS is prone to attacks, such as:

- *Eavesdropping and hijacking:* Instigated as MitM, Relay, Advanced Persistent Threat (APT), Sybil, and Spoofing attempts [95], wireless communication channels are being hijacked to retrieve information transmitted in this way. The MitM attacks are plausible between the 3G network and the non-3GPP WLAN networks that compromise the internal virtualization infrastructure entities of the edge level [7]. The lesser level of encryption and integrity in low-resource IoT devices poses higher risks of compromising these channels that connect to the MEC edge system [96]. These attacks would gain access to the ME Apps operating in MEHs and manipulate the virtualization infrastructure of the MEH to exhaust its resources while infecting other MEHs connected to the infiltrated one. Thus, a privileged escalation attempt could be launched via these attacks.

- *Jamming and Denial of Service (DoS):* The purpose of jamming the wireless channels and disrupting a service via DoS attacks is to violate the availability aspect of the RAN connected to the MEC edge system. These attacks pose a higher destructiveness towards novel systems due to the delay they cause for latency-tolerant applications. Thus, with a compromised MEH, services to the consumers can be blocked or the MEC system-level entities can be disrupted through unnecessary queries to impede the service of the entire infrastructure. Novel botnet-type DDoS attacks are capable of challenging the access capacity of

UALCMP and CFSP entities at the MEC system level.

- *Malicious node injection:* The diversity of the UEs connecting to the BSs would present an issue of managing the compatibility among mobile devices produced by different vendors and the communication protocols of varied UEs or UE Apps. In such circumstances, malicious nodes could be injected into the system, exploiting the vulnerabilities in the UE devices: these include susceptibility to device cloning, less secure wireless protocols (WPA/WPA2), proneness for hardware Trojans, predictable access control credentials (PIN/ pattern), or alleviated resiliency against malicious software agents disguised as partial entities of UE Apps such as spyware, adware, Trojans and malware. Such vulnerabilities could feed malicious content or counterfeited information to the ME Apps through the BSs, which would manipulate the services offered by the MEH, as mentioned in TV E3. However, these attacks could be mitigated by employing an effective authentication mechanism embedded with a trust verification scheme for the UE and BS connectivity.

Summary: The usage of conventional security primitives, though they are enhanced, is an aggregated unwanted burden on the application layer payload space of the traffic delivery. Despite the improved BW and multiple channel support furnished by 5G and MEC, these cumbersome approaches are limiting the extent of novel applications, such as autonomous vehicles, to broaden their features. Thus, lightweight primitives as indicated in the study by Chen [97] or QR crypto approaches [88] can be engaged to heighten the complexity of the security schemes. Another way to reduce the load of the application layer is to embed the security mechanisms into the PHY layer as in the PLS approaches specified above. However, these approaches are diverse and reliant on the communication device (i.e. vendor architecture), medium (i.e. wireless/ wired, FO), and technology (i.e. BLE, Wi-Fi, Zigbee, or LoRaWAN). Therefore, compliance with the PLS primitives to be employed should be established to prevent interoperability and compatibility issues with PHY layer protocols. 5G-based RF networks are still in the experimental stage. Both channel models and network architectures should be specified and standardized for each 5G-

based use case to avoid discrepancies after deployment. Security and privacy should also be considered primary requirements when forming such standardization.

3.1.2.1.2 A2 - Ad-hoc connectivity between User Equipment The threats on A2 are associated with the links that are established between UEs in an ad-hoc manner. These links employ short-range communication channels that are used for data-transferring purposes under the influence of specific UE Apps. The connectivity type is Device-to-Device (D2D) which establishes a direct communication link between two devices, without requiring any BS for connectivity [98]. Short-range communication technologies such as Bluetooth, Bluetooth Low Energy (BLE), Near Field Communication (NFC), ZigBee, Wi-Fi direct, narrowband IoT (NB-IoT), SIGFOX, or any technology which could form a Mobile Ad hoc Network (MANET) are capable of deploying connections between UEs [74, 7, 37]. Moreover, FlashLinQ and Proximity Services (ProSe) are also capable of forming D2D communication platforms. FlashLinQ, developed by Qualcomm facilitates content sharing, gaming, and social networking features for proximity devices. The ProSe is a standardization published by the 3GPP for enabling proximity discovery and direct communication for future AN-based deployments [98].

Vulnerabilities: The vulnerabilities of this TV are limited to the communication channels established between the UEs. The threats originating in a UE do not directly influence the intrusions into MEC systems in this TV. A UE infiltrated by a D2D-based attack could use its connectivity with the BS to infect the MEC servers for various manipulations of the MEH explicated in E3.

- *Attacks on short-range communication technologies:* Attacks such as eavesdropping, impersonation, forging, free-riding, DoS, and privacy violation are probable [98]. Most such attacks are feasible due to the nature of the communication protocols embedded within short-range communication technologies. These technologies prioritize leveraging the bandwidth for D2D execution rather than employing security measures.
- *D2D traffic offloading:* The method of offloading cellular traffic to the

UEs by the MNO is an example of a D2D instance [98]. In this approach, MNO only transmits content to specific UEs considered as cluster heads, and those UEs are multi-casting the content to respective UEs in the scope of the cluster. Moreover, use cases, such as extending the coverage through D2D connectivity for rural areas and establishment of critical communication channels for disaster or terrorist situations (where the cellular network is disabled), envision the future potential for D2D-based services. In these use cases, connectivity between a UE cluster head and the MEC servers is maintained for content sharing under the supervision of the MNO. Thus, this scenario opens up new possibilities for adversaries to exploit the cluster head UEs for manipulating the service offered by them.

Summary: The main drawback of D2D communication in terms of security is the resource scarcity attributed to the IoT and CPS devices. Thus, lightweight approaches are essential to conserving energy, while security keys, hashes, and authentication codes should be generated in an optimal way. As these protocols are mostly autonomous, authentication credentials are computed in an algorithmic manner that can be replicated by a resourceful adversary. Thus, PUF fulfills a lacking aspect of M2M communication by employing unique and non-crypt analytic parameters to secure the D2D channels. However, in authentication stages or in a layered security circumstance, the repeated message flows included with encapsulation, coding, and modulation constructs consume energy that does not contribute to the throughput. Therefore, selective minimal security features/mechanisms should be identified for each D2D or M2M function to maximize the operating time.

3.1.2.1.3 A3 - User Equipment (UE) UE can be a mobile, personal computer, CCTV camera, or wearable sensor or sensory system which can be in direct contact or connected through a gateway device to the BS. The variety of technologies attributed to a UE on the aspects of operating systems (Android, iOS, Windows, Symbian, BlackBerry, and WebOS), memory management (SD, micro-SD, and HDD), communication (RF, RFID, NFC, Blue-

tooth, Wi-Fi, and Ethernet), physical design and structure contribute to the improbable deployment of a generic security solution for UEs in a holistic extent. The UE contains information related to various aspects of the daily life of a person, such as private information (photos, medical reports, medical statistics, and CCTV footage), location (GPS), daily routines (shopping and transportation), enterprise information, critical infrastructure information (energy consumption, financial, banking, and emergency service status) and online account statistics—where divulging such credentials and parameters could be fatal for one's well-being [37]. Thus, threats to mobile users' privacy are of great concern [99],[18]. The resources embedded in a UE in terms of processing power, storage capacity, and battery life are the most significant factors for this threat vector [96]. Certain softwareized and virtualized attacks require a minimum level of resources to launch in an executable platform. Thus, enhanced processing and storage resources in the current UEs improve the possibility of launching such attacks that are capable of hindering detection by conventional means.

Vulnerabilities: The threats could be instantiated by a UE with or without the knowledge of the user. Even a genuine user is capable of activating a malicious software agent unintentionally. This risk of UEs being vulnerable to both physical and remote attacks makes this threat vector extremely critical. The UEs are vulnerable to physical damage, Side Channel Attacks (SCA), malicious code injection, and hardware Trojans, while all other attacks explicated in TV A1 and TV A2 are applicable to the communication interfaces.

- *Physical attacks:* Physical damages are the most common type of attack for this TV, where they lead the attacker to re-configure the affected device such that they convey misdirecting information to ME Apps [18]. These misinforming attacks lead the MEC system to interrupt its services by feeding fake but calculated information to edge devices. A typical scenario would be the reconfiguration of ME Apps to execute continuously (without termination and commandeering maximum resources) and exhausting its resources.

- *Side channel attacks:* The attacker's intention in launching an SCA is to

extract the cryptographic parameters by cryptanalytic means. Acoustic cryptanalysis, electromagnetic analysis, timing, power monitoring, and differential fault analysis are such SCAs applicable for UEs [100]. Security protocols engaged in communication channels are exposed with such revealed credentials. Since these attacks are arduous to detect due to their variety, countering consumes time and resources.

- *Mobile delegation:* The offloading of services due to mobile delegation could result in UE experiencing various offloading mechanisms such as full offloading and partial offloading. These offloading mechanisms are prone to attacks [101]. An infected UE App, or a UE when offloading partial or complete executable, or when offloading passive content to the MEC server has the ability to inject a malicious agent to be activated in the corresponding ME App in the MEH. Such an attack directly contributes to the TVs E3 and E4.

- *Vulnerabilities in the gateway devices and channels:* In the case of UE acting as a Machine Type Communication Gateway (MTCG) for applications like e-health or any other MTC deployments, the malicious content could be generated at one of the sensors or actuators where the UE is acting as an intermediary ingress point to the intrusion. Further, the communication channels (i.e., mostly the over-the-air type) are subjected to the same kind of threats, where malicious agents can be instilled in addition to the generic intrusion-based threats that violate CIA aspects. Such injected malware or viruses pose a high-risk factor for UEs, as their means of penetration can occur in various ways. The repercussions of a malware attack resemble a malicious code injection attack. However, the damage level it causes is reliant on the malware type—for the variants of Trojans, worms, rootkits, Spyware, Ransomware, and Adware.

- *Overloading resources:* Resource allocation and scheduling of the ME App is controlled by the Mobile Edge Platform (MEP) in each MEH, where the MEP communicates with the UE App in case of a mobile delegation circumstance [41][101]. An infected UE App could influence the

MEP to allocate inessential resources at the MEH, causing a service interruption. The capability of UE to be deployed in a tamper-resistant manner, employing lightweight but effective cryptographic primitives with high resiliency relies on the design and the manufacturer of the device.

Summary: The higher resources and functions available for mobile devices are attracting novel, application-level threats, in addition to the typical malware, SCA, physical, or cloning attacks. In fact, adversaries can combine the method of attacking, where both malware and SCA-type attacks can be perpetrated in a single threat attempt. A typical IDS is not adequate to detect all such novel attacks. Thus, application-level security features should be embedded into mobile devices in their design stages to detect and prevent them. Though there are techniques for detecting SCAs currently, novel side channels are determined by adversaries from time to time. The processors, circuitry, and TRX interfaces should be subject to extensive security tests prior to releasing the product.

TABLE 3.2 summarizes the countermeasures and the best practices for mitigating threats from within AN-based threat vectors.

3.1.2.2 Threat Vectors related to the Mobile Edge Network (MEN)

The entities located in the edge network or the host level of the MEC paradigm, such as the Mobile Edge Platform Manager (MEPM), Virtualization Infrastructure Manager (VIM), and the Mobile Edge Hosts (MEHs), are investigated in this section for establishing the threat boundaries. The placement of MEC servers is localized in comparison to conventional data centers. Thus, the MEC edge (i.e., host) level is prone to tangible or physical security attacks [1]. These TVs are depicted in Fig. 3.3.

3.1.2.2.1 E1 - Mobile Edge Platform Manager (MEPM) The MEPM is the entity that monitors the MEH activities, using the connectivity that it maintains with the Mobile Edge Platform (MEP) contained by the MEH. This entity performs resource allocation and monitoring functions with the connections that it maintains with VIM, MEO, and OSS. As MEPM is the highest level entity in the

Table 3.2: Summary of Countermeasures for Threat Vectors in Access Network

Ref. No.	Proposed Countermeasures	TVs in AN		
		A1	A2	A3
[102]	Encrypting payload with AES 256-bit and securing signaling with OWS	✓		
[90]	5G wireless security architecture	✓		
[103]	PLS model for multi-tier HCN	✓		
[104]	RT based channel model for 5G mmWave small cell	✓		
[102]	Two-way mobile number based D2D authentication scheme		✓	
[98]	Layered security deployment		✓	
[105]	E2E authentication scheme using IBE PHY-ID		✓	
[106]	PUF scheme for FPGA based on TERO		✓	
[107]	PUF based Two factor authentication scheme for mobile phones		✓	
[108]	Anomalous detection using machine learning			✓
[109]	SPE framework for UEs and intent-based validation policy			✓

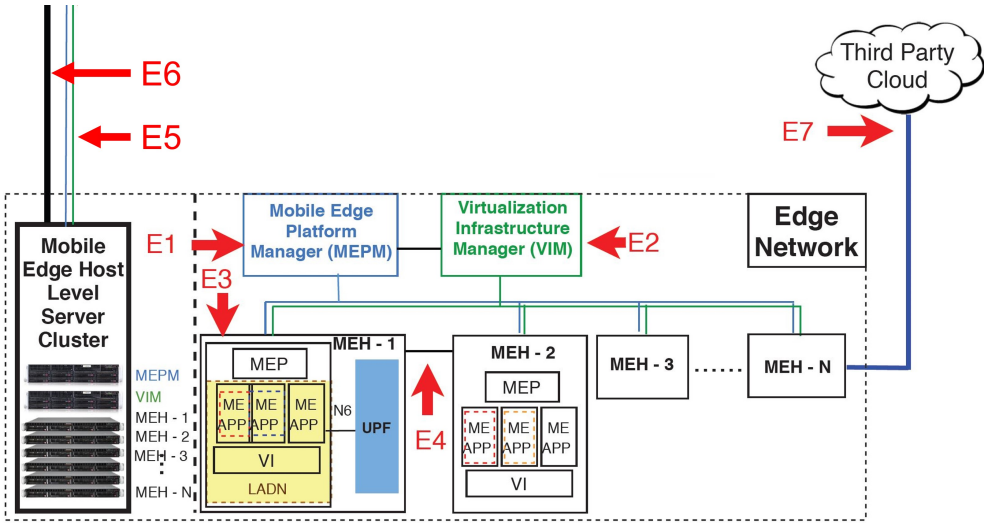


Figure 3.3: Threat Vectors in the MEC Edge Network.

MEC host level, it is responsible for performing the following functions: managing the ME App life-cycle, traffic steering, and recording fault and performance

metrics from VIM. Moreover, MEPM reports the holistic monitoring statistics of the host-level entities to the system level.

Vulnerabilities: The placement of MEN entities at the edge limits the physical tampering-based attack vectors. However, the risk is higher compared to conventional CC.

- *Feeding fake configuration/ feedback data:* MEPM could be furnished with fake information regarding the resource allocation–based configuration or the feedback data. Such a threat could originate either within the AN or from an infiltrated ME App. As the MEPM is connected to the OSS, MEO, and VIM, such false information is disseminated to the system-level entities, destabilizing the entire MEC system.

- *Infected through ME Apps and UE Apps:* An infected ME App could force the corresponding MEP to lead the MEPM to allocate undesired resources to induce service disruption. An already exposed UE App or a communication channel in the AN has the capability to mislead the MEPM such that it allocates more MEHs for processing, leading to resource depletion.

- *VM based attacks:* VM-based attacks such as VM manipulation, Domain Name System (DNS) amplification, VM escape, Virtual Network Function (VNF) location shift, and security log troubleshooting attacks are probable for this threat vector.

3.1.2.2.2 E2 - Virtualization Infrastructure Manager (VIM) VIM executes the integral task of facilitating ME Apps of virtualized infrastructure resources in every MEH within a single edge vicinity. The connections of VIM towards MEO and MEPM feed the statistics to perform the tasks of managing and monitoring the Local Area Data Network (LADN) deployed in MEHs. As this is the main entity assigned for facilitating the virtual resources at the edge, the role of the VIM is similar to a hypervisor for MEHs. Thus, VIM performs the functions of allocating, managing, releasing, and performance monitoring virtualized resources [41].

Vulnerabilities: The hypervisor functionality of the VIM attracts adversaries whose intention is to manipulate resource allocation capabilities, targeting resource depletion.

- *VM based attacks:* As VIM is responsible for allocating resources, it could be subject to attacks such as VM manipulation, VM escape, or any malicious attacks targeted for virtual deployments. These attacks would exhaust the system resources via various methods, such as allocation of inessential processing and storage facilities for a single ME App in the Virtualization Infrastructure (VI), allocating excessive amounts of ME Apps to process a single application, or blocking resources or interrupting services to a particular ME App.
- *Misleading system level entities :* The system-level entities could be confiscated for privilege escalation or service interruption threats due to their connections with VIM. If the VIM is compromised by an attack, malicious misconfiguration exploits could be launched by an attacker [89].

3.1.2.2.3 E3 - Mobile Edge Host (MEH) MEH is the main host-level functional entity that performs the computational, storage, or networking operations in the MEC paradigm. A MEH consists of an MEP, Virtualization Infrastructure (VI), and a data plane or Local Area Data Network (LADN) which maintains the local connectivity among ME Apps. Additionally, the User Plane Function (UPF) is a 5G access network entity included inside the MEH for integrating the 5G core network into the LADN. As MEH is the only entity that stores the content conveyed from UE Apps, the risk of being exploited is high.

Vulnerabilities: MEHs are the main target of any attack originating from the AN towards the MEC system. As they are the main functional elements that support service processing, storage, and computation, any attempt to penetrate the MEC system through a malicious act should be directed toward the MEH from the attackers' point of view.

- *Computational offloading:* The threat of a MEH being subject to an infection by a malicious adversary is highly reliant on computational offloading processes, as discussed in the AN threat vectors. Once infiltrated, it

has the capability to mislead the MEP and VI for resource allocation and service continuation.

- *VM based attacks:* The attacks applicable to any VM-based deployments are prone to ME Apps, as they are deployed in a VI. As MEHs are launched as VMs, all such attacks are directly impactful to its operation. Furthermore, service-impeding attacks such as DoS or DDoS affect the autonomous operation of the MEHs.

- *Feeding false statistics to exploit internal entities:* The false statistics conveyed by the affected ME Apps could cause misconfigurations in the VI and the MEP, which could be exploitable by a privilege escalation-type attack. These could lead to service disruption through resource depletion.

- *Exploiting the connection to the UPF:* The UPF component included in the MEH maintains a link to the 5G core network. An attacker can exploit this link to attain networking credentials. Further, an infected MEH can feed false information to the core network and compromise the stability within core network entities.

- *SCAs on VMs:* Shared memory–based, cross-VM, cache-based, and energy consumption–based SCAs are possible for VM manipulation [110].

3.1.2.2.4 E4 - Connectivity between Mobile Edge Hosts ME Apps might require connecting with one or several MEHs for processing high-end applications—in which a single MEH does not possess the resources to perform the intended function. This requires the connectivity between ME Apps operated under MEHs, which are established through MEPs. This is probable for high-end applications such as Industrial IoT (IIoT), surveillance, or critical infrastructure services. As the VI and LADN of two MEH entities are accessed, the respective VIM and UPF should be notified in addition to the subscriptions in the MEPM.

Vulnerabilities: A malicious ME App emerged from the methods explained under E3 TV and is capable of infecting other connected ME Apps and MEP elements, in addition to the entities inside a MEH. However, the connections among MEHs are internal and obscured to adversaries. Thus, MitM-type attacks are improbable.

- *Malicious injections:* A malicious injection occurring in the AN, after traversal into a MEH, is capable of infecting another MEH through the E4 connection.
- *Rouge ME Apps:* An infected ME App could manipulate the MEPM for resource depletion while depleting the resources of each MEH that a malicious agent manages to propagate.

3.1.2.2.5 E5 - MEC platform connectivity between the edge and the core

This bi-directional connection between the mobile edge system-level entities and the host-level entities is a critical link in the MEC paradigm. As these two levels are separated by the location, the connectivity could be established from long-range communication links using technologies such as Microwave (MW), Fiber Optic (FO), Satellite, or RF. The registration process for a particular MES requested by a UE App is established through this link [41]. UEs connecting to a MEC system should first register at OSS through the connection extended from the BS to the UALCMP entity in the core [111]. This is the initial interfacing of UE Apps with the MEC platform. The nature of the UE App (whether it is operated by a trusted or a malicious entity), authenticity, and content transmitted from the UE App to the ME App are factors to be investigated in this threat vector.

Vulnerabilities: A resourceful adversary could manage to intervene at this communication link. Even though the possibility of intervention is lower with long-range communication links, the exposure of the control information could give the attacker the opportunity to exploit the MEPM, VIM, and MEH host-level entities as desired. Similarly, if the attacker managed to alter the status-updating parameters conveyed from the host level to the system level, MEO and OSS could be subject to service interruption attacks.

- *Attacks on radio channels:* RF links are vulnerable to attack vectors discussed under A1.
- *Attacks on MW links:* Electromagnetic Pulse (EMP) based, Sybil, DoS, and DDoS attacks are probable with MW links [112].
- *Attacks on FO connections:* FOs are vulnerable against fiber tapping and hidden pulse attacks [113].
- *Attacks on satellite links:* Satellite Communication (SATCOM) links face the threats of kinetic, jamming, and cyber-attacks [114].

3.1.2.2.6 E6 - Connectivity between Mobile Edge Apps operated under Mobile Edge Hosts at different Base Stations In this situation, the UE App related to the user accesses ME Apps operated under MEHs at different mobile edge host levels residing in two BS locations controlled under a single MEO (or ME system level). A crowd-sourcing application or a smart grid application that deploys two instances of the same ME App operating at different locations is an example of such a scenario. Even as ME App instances operate at geographically dispersed edge levels, they are governed by a singular trust domain contrived by an OSS and an MEO.

Vulnerabilities: The connectivity between two host levels governed by the same system level could be prone to intervening attacks as explicated in E5. These interposing attacks are capable of penetrating both the MEC host levels.

- *Intervening attacks:* Attack vectors perpetrated as MitM or Relay can be applied to various communication link types, as presented in E5.
- *Attacks on migrating services:* The service migration from one host level to the other poses the possibility of infecting two host levels through malicious content. As these applications are only probable with upscale use cases, the effects of the attacks could be incisive.

3.1.2.2.7 E7 - Connectivity with the Mobile Edge Host and the Cloud Servers In the case of services hosted by third-party consumers, MEH main-

tains a connection to a centralized cloud or a server platform as a typical massive IoT-based cloud implementation. In that scenario, the MEH (or MEHs) pre-process the content in their possession before conveying them to the cloud service. Typically, the cloud service hosts the edge services as a Function as a Service (FaaS) implemented through a cloud wrapper MEC application [42]. The connection to the cloud server is instigated through the LADN of the MEH.

Vulnerabilities: The security policies and rules adopted for this type of channel are a rarely investigated area. CSPs are susceptible to such attack vectors. Thus, interoperability issues might be plausible.

- *Intervening attacks:* Intervening attacks such as MitM, relay, or impersonation attacks are plausible for the communication channel extending from MEH to the cloud platform. Masquerading attacks launched by adversaries to appear as a cloud service can perpetrate sinkhole or wormhole attacks.

- *Packet sniffing attacks:* The channel between the cloud service and the MEH is vulnerable to traffic sniffing attacks for perpetrating geo-location leakage or any other data exfiltration attempt [18].

- *Malicious injections:* An attack launching from this channel, due to its bi-directional connectivity and exposed nature, could result in a malicious agent in the MEH, enabling E3-based threats. Moreover, the cloud platform or centralized servers are prone to malicious attacks from an infected MEH.

Summary: Determining the novel threats possible on an edge platform formed with virtualization technologies is a challenge that should be addressed prior to deploying MEC. TPMs offer the ability to attest the connecting UEs to determine their legitimacy. This is an important fact that leads to preventing the edge platform from malicious penetrations. Furthermore, VMIs are key tools for determining the anomalous behavior of virtual entities. These two technologies together form a protective shell to prevent malicious injections from ingressing to the MEC edge level. A VMI can be attached to the VIM that monitors the VM performance, while a TPM can be connected with the MEPM

for distinguishing malicious ME Apps from legitimate ones. In addition, security mechanisms can be embedded into the virtual infrastructure when forming virtual entities or VMs. However, lightweight virtualization or containerization technologies are attributing lesser securing mechanisms. Thus, security constructs should be implemented at the VM or hypervisor platform level. Furthermore, the orchestration function of the MEPM should be designed with security-aware features.

TABLE 3.3 summarizes the countermeasures and best practices focused on MEN-based threat vectors.

Table 3.3: Summary of countermeasures / best practices for Threat Vectors in Mobile Edge Networks.

Ref. No.	Proposed Countermeasures / Best Practices	Applicable TVs in MEN						
		E1	E2	E3	E4	E5	E6	E7
[89][115]	TPM for validating resource exhaustion	✓	✓					
[89][116]	Form DMZs to apply access control and firewall policies at VI	✓	✓	✓	✓	✓	✓	✓
[89]	Hypervisor introspection tools serving as a HIDS	✓	✓	✓				
[117]	Policy based VMI IDS framework	✓	✓	✓				
[89]	Encrypting VNF Hard disks	✓	✓	✓				
[89]	Signing VNF images	✓	✓	✓				
[89][115]	Using a remote attestation server	✓	✓	✓				
[118]	Security framework for SDN/NFV deployments in IoT	✓	✓	✓	✓	✓	✓	
[119]	On-demand dynamic SFC based security service model	✓	✓	✓	✓	✓	✓	✓

3.1.2.3 Threat Vectors related to the Core Network

The core network expands from the MEC system-level devices such as UE App Life-cycle Management Proxy (UALCMP), Customer Facing Service Portal (CFSP), Operations Support System (OSS), and Mobile Edge Orchestrator (MEO) to the backhaul network that extends to the Internet connectivity. Fig. 3.4 illustrates the locational context of the discussed TVs.

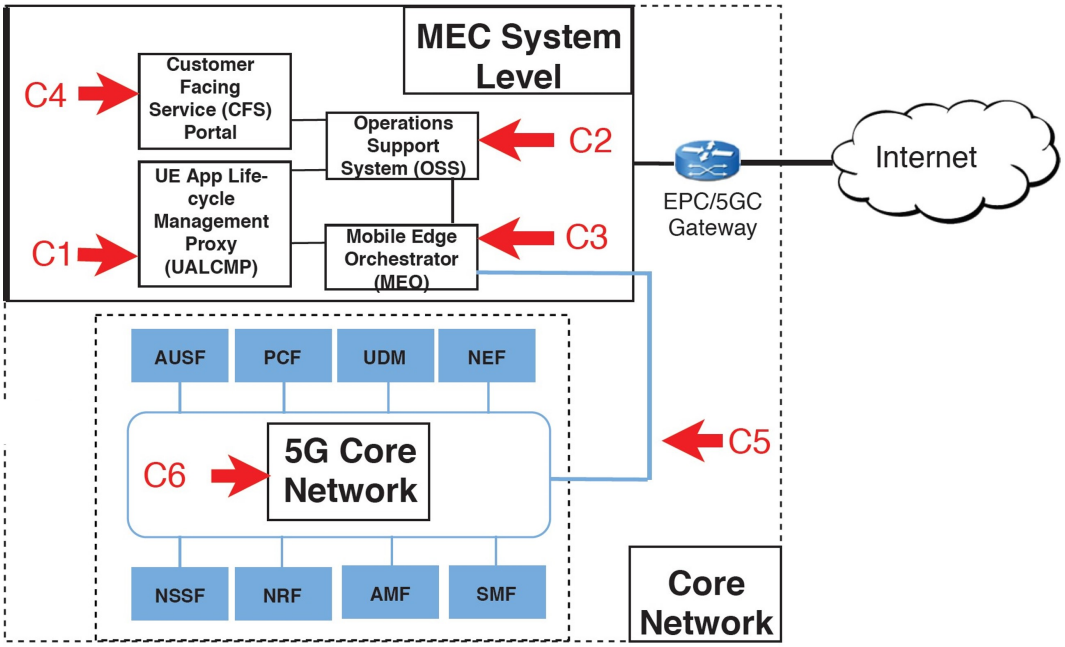


Figure 3.4: Threat Vectors in the MEC Core Network.

3.1.2.3.1 C1 - User Application Life-cycle Management Proxy (UALCMP)

UALCMP is the initial contact point for any UE App that intends to subscribe to MEC services. Its function is handling multiple UE App requests while determining their life cycle. As this entity includes proxy functionality, the internal addressing function is facilitated for the MEC system to link UE Apps to their corresponding ME App or ME Apps operated at the MEN.

Vulnerabilities: The attacks perpetrated on the UALCMP are targeted at its access interfaces for overburdening them.

- *Attack vectors on the access interface:* The request handling nature of this entity has the possibility of DoS, DDoS, or masquerading attacks; these would entail service disruption or access granting for malicious intruders.

- *Manipulating life-cycle of Apps:* As the UE App life-cycle is determined by this entity, an adversary is capable of furnishing falsified information for obtaining an increased life-cycle beyond its requirements.

- *Consequences for the OSS:* As the OSS is dependent on the requests and information of the UALCMP, the service disruption of UALCMP could directly affect the OSS operations.

3.1.2.3.2 C2 - Operation Support System (OSS) The OSS grants the service requests forwarded through the UALCMP or CFS portal while instantiating or terminating ME App functions. Additionally, OSS maintains links to MEPM and MEO for extracting control information. OSS grants the approval for subscribers to use ME Apps that are configured for a particular MEC service. Thus, this entity is critical to the attackers for gaining access to the MEC system.

Vulnerabilities: As the MEC host level is reliant on OSSs' approval for instigating the MESs, the attacker's intention is to delay its operation through the UALCMP.

- *Service denying attacks:* As the UALCMP is the entity facing the service requests from UE Apps, the OSS has to be protected from DoS or DDoS attacks.

- *Feeding false information in the registration process:* Since all the UE Apps subscribing to Apps should be registered in the OSS, the attackers could attempt to inject fake information to impersonate valid entities for pertaining MEC services. If an attacker were successful, the mobile delegation–based operations would enable the possibility to infiltrate the MEC host level.

3.1.2.3.3 C3 - Mobile Edge Orchestrator (MEO) MEO represents the core functionality of the MEC concept, which assigns the role of the hypervisor for the holistic MEC system. It observes the deployed MEC hosts and resource utilization status at the edge. As the hypervisor, MEO supervises the VMs and underlying hardware configured for virtualization [89]. The main functions of MEO are service migration, mobility management, and traffic steering monitoring. The hypervisor role of MEO is still applicable to scenarios where the MEC system is integrated with other driving technologies such as NFV Infrastructure (NFVI) with NFV Orchestration (NFVO) capability [53].

Vulnerabilities: In the scenario of the MEO acting as the hypervisor being compromised, the automated network configuration exploits, orchestration exploits, malicious misconfiguration, and SDN controller exploits are probable [89]. The configuration parameters forwarded from the entities such as MEPM, OSS, UALCMP, and VI are also vital to the utilization of the MEO. Therefore, mechanisms should be employed for detecting such attacks and remedying them.

- *Resource manipulation attacks:* Though MEO is deployed at the system level where malicious intrusions are improbable, resource allocation and service manipulation attacks such as DNS amplification and VM escape would be highly probable. Due to the effects of such attacks, the configuration of the MEO system could be destabilizing such that it cannot perform at its optimal level.
- *Security-log troubleshooting attacks:* As in the case of a security log troubleshooting attack, the logs of the operations of MEO or any other entity would be altered. Thus, the control statistic could not be conveyed to the corresponding ME entities for optimal operation. Even if the log information is conveyed, the entities would malfunction due to their altered content.

3.1.2.3.4 C4 - Customer Facing Service Portal (CFSP) CFSP facilitates the access of ME Apps to third-party services, where it is capable of recalling service-level information from such applications [41]. Car park monitoring,

connected vehicles, and IoT big data are applications suited for MEC deployment. These deployments use sensors that gather enormous amounts of data, which are pre-processed at a MEC edge server and conveyed to a centralized corporate server for further analysis [101]. For most of these services, MEC acts as a low-latency aggregation point. Thus, third-party consumers instigate the service requests from Cloud Service Providers' (CSPs) end. The role of the CFSP is to approve the deployment of MEC resources for the processing of such third-party requests.

Vulnerabilities: As the role of the CFSP is handling requests, it could be prone to service-based attacks such as DoS and DDoS. Moreover, proper approval mechanisms should be employed by the CFSP to enable traffic steering of third-party applications to the MEC host-level entities. Use cases applicable to E7 are examples of extended services approved by the CFSP for third-party applications. A compromised CFSP could manipulate the service subscriptions of OSS.

3.1.2.3.5 C5 - Connectivity of the Mobile Edge Orchestrator and the 5G Core Network A secure interface has to be defined for the MEO and 5G core network[120]. The control signals will be exchanged between the 5G core network and MEO via this interface. Since this is the main interface that interacts with the 5G core network, this is one of the critical interfaces in the whole MEC architecture. It is possible to host both 5G core network elements and MEO in the same physical host. However, it is more likely to implement them in two different physical hosts [121]. In that case, the communication link should be established between MEO and 5G core network via the physical network. This network can be implemented using any network technology such as wireless, wired, or optical. Depending on the connecting medium and the deployed location, these entities will face different security challenges.

Vulnerabilities: The interface between MEO and 5G core is yet to be defined. However, several security threats can already be identified relative to this interface.

- *TCP/IP attacks on the channel linking MEO and the 5G core:* It is more

common to use separate physical hosts for 5G core network and MEO [121]. In that case, the control traffic will be transferred between two entities via an open 5G backhaul network. Therefore, the interface between MEO and 5G core will be vulnerable to typical TCP/IP attacks such as eavesdropping, spoofing, DoS, replay, and reset attacks. It is mandatory to enable proper security mechanisms, such as mutual authentication, E2E encryption, or Challenge-Response Procedures (CRPs), to mitigate these issues. The impact of these attacks will be minimum if both MEO and 5G core are deployed in the same physical host.

- *Lack of a standard interface:* Another challenge is to properly define an interface between 5G Core and MEO[122]. It is challenging to define a proper and unified security mechanism without a standard interface. However, this challenge is somewhat relaxed since ETSI[123] is leading both 5G and MEC standardization tasks.

3.1.2.3.6 C6 - 5G Core Network Ultimately, the 5G core network controls the entire 5G network. The 5G core network will enable the MEC capabilities for the selected services. Moreover, all the control signals will be forwarded to the MEC system via the 5G core network. Therefore, the 5G core network is the vital element ensuring the proper operation of the whole MEC system.

Vulnerabilities: Since the 5G core network is the main control entity of the whole 5G network, any attack on the 5G core network will have a significant impact on the 5G MEC system.

- *Nature of the softwarized core:* In contrast to pre-5G networks, 5G networks have a softwarized or virtualized core [124]. Here, all the core network functions are implemented as VNFs. However, several security concerns are observable in the functionality of VNFs. The hardware-based pre-5G core network had natural protection against many attacks due to its closed, complex, and vendor-specific nature [125]. However, the NFV base 5G core network is open and software controllable. It is comparably easy to manipulate a software-based system than a hardware-based system.

- *Typical VNF based attacks:* VNFs are vulnerable to attacks such as interoperability issues [126], VM escape [127], VNF Manipulation [89] and VNF location shift attacks [89].
- *Mismatching policies:* Different VNFs are developed by different VNF providers, and they attribute different levels of security policies. The mismatch between these differences can lead to vulnerabilities when they are deployed in the same system [126].
- *VNF based service denying attacks:* A variety of DoS/DDoS attacks on targeted services is possible when VNFs are hosted in the cloud, e.g attack on Bitbucket [128]. The impact of DDoS is even greater for virtualized networks since this attack could spread to untargeted VNFs that are hosted on the same physical host [129].
- *VNF software flaws:* Since VNFs are software, they are vulnerable to software flaws which can lead to unintended behavior. For instance, these software flaws can be used to bypass firewall restrictions or perpetrate buffer overflow to execute arbitrary code [129].
- *Hypervisor flaws:* A malicious VM can escape from the virtualization environment and execute arbitrary code within the hypervisor to compromise it [127]. An attacker misuses the privileges of a compromised hypervisor to install kernel rootkits in VNF's OS and to manipulate the VNF [89].
- *Issues in migration:* An attacker can migrate from a compromised VNF to a different location where fewer security or privacy policies are enforced to gain additional access to the system [89].

Summary: The UALCMP and CFSP are the interfacing entities at the MEC system level. These two entities can be subjected to DoS-type attacks. In order to mitigate such attacks, an attestation server can be employed for approving the requesting UE Apps. Hence, a trusted domain can be established and centered around the OSS that contrives an NFVI trust framework with TPMs at different levels. The MEC system level will be developed in a

virtual environment. Thus, kernel hardening tools would protect the MEC entities at the operating system level. Further, hypervisor introspection is a key requirement for the system-level entities to monitor anomalous processes occurring at both edge and system levels. Such a function can be embedded as a construct into the MEO to provide a holistic overview of malicious patterns. In addition, an agent of the said security construct should be deployed at the edge, connected to the MEPM to perform securing acts. The 5G core network standardization and its integration into the MEC system level is still a grey area. Thus, the interrelations of MEC system-level entities to the 5G core network entities should be standardized in the near future.

Table 3.4 summarizes the countermeasures and best practices for mitigating threats originating within the core network-based threat vectors.

3.2 AR - Service Migration

Service migration is the process of transferring executable content configured to offer a specific MES, either between edge levels or between the cloud and the edge [137]. This process could expose unprecedented vulnerabilities and flaws in a MEC environment. In a CC-based service migration, services originally hosted in cloud environments are migrated to the edge servers located proximate to the mobile devices. This reduces latency and improves the capacity of the access network. Thus, as the services are executable programs, tools, or software running on a virtualized platform, the code of that particular software should be migrated to the edge in such circumstances. There are four approaches being considered for code migration by Rodrigo et al. [37]: 1) migrating only part of the code; 2) migrating an exact replica or a clone of the entire execution environment with the memory and CPU images; 3) migrating mobile agents created by mobile devices to the edge; and 4) amalgamating process cloning and mobile agents at the edge. The MEC services are typically launched as VMs in the VI of MEHs. VM migration is conducted as either live or non-live approaches [138]. In a non-live migration, the entire VM with its running states is encapsulated and transferred to the migrating vicinity, while

Table 3.4: Summary of Countermeasures for Threat Vectors in the Core Network

Ref. No.	Proposed Countermeasures / Best Practices	Applicable TVs in CN					
		C1	C2	C3	C4	C5	C6
[89]	Updating the security patches timely			✓			
	Limiting the operational time of remote access services only when required						✓
	Employing strong password policy			✓			
	Using SELinux kernel and its tools	✓	✓	✓	✓	✓	✓
[89][117]	Hypervisor introspection			✓			
[115]	Linking remote attestation with host and system levels	✓	✓	✓			
[118]	Security framework for SDN/NFV deployments in IoT	✓	✓	✓			
[130]	A framework to apply adaptive trust evaluation and sustainable trusted computing technologies to ensure computing platform trust and achieve software-defined network security	✓	✓	✓			
[129]	Discuss the security issues in SDNs when virtualized as VNFs		✓	✓		✓	✓
[131]	Study the feasibility of extending the current NFV orchestrator to have the capability of managing security mechanisms					✓	✓
[91]	Propose a security orchestrator apply to security management in ETSI NFV architecture	✓		✓		✓	✓
[132]	Present a threat analysis and corresponding security requirements in the context of NFV	✓	✓	✓		✓	✓
[133]	Analyze the challenges on Datacenter in the form of Network Security Function Virtualization (NSFV) over Openflow infrastructure					✓	
[134]	Present the different architectural design patterns for the integration of SDN/NFV-based security solutions into enterprise networks					✓	✓
[135]	Present the integration approaches of network and security policy management into the NFV framework			✓		✓	✓
[136]	Provides a method of identifying the first hardware unit attacked by the security attack and security mechanism for NFV-based communication networks			✓		✓	✓

the local operation suspends completely. In live migration, VMs are orchestrated simultaneously at different edge platforms without suspension, while

multiple VM migrations are plausible via Local Area Network (LAN) or Wide Area Network (WAN) coverage. Other than services, migrating computational processes are viable applications for MEC deployments where the network controller acts as the resource selector for utilizing the computation power. The computational migration models could be formulated using the Markov Decision Process (MDP) problem based on a random-walk mobility model or a threshold-based model such as the Lyapunov optimization technique [1, 99].

Vulnerabilities: The migration of services means migrating an entire serviceable platform or a part of it to the mobile edge hosts operating at the edge. Still, the security of the migration process is a grey area due to the diversity of utilized resources and the scope of the services. The migration process begins with a service operated within a single MEH or multiple MEHs. The unauthentic nature of the Internet-based connectivity among MEC edge entities poses security issues extending to VIM, UEs, and the migration data traversing channels [138].

- *Malicious code injection:* Malicious code injection attacks targeting the migration channels are imminent at the edge network. Detecting the malicious code would be improbable once the migration process is completed. This leads to the exploitation of communication links between the edge service infrastructure and the cloud server or core network entities.

- *Attacks on mobile agents when migrating:* In the case of employing mobile agents for service migration at the edge, the probability of an intrusion is higher as the code or the service platform migrated from the access network is subjected to threats under A1. Thus, it is imperative to secure the migration processes with proper security mechanisms for mitigating massive service manipulations at the edge of the MEC deployment regardless of the diversity of the services.

Existing Solutions: Understanding the dynamics involved in service migration is critical to developing security measures for the MEC architecture.

- *Secure migration framework:* Machen et al. [139] introduced a layered framework for migrating active services using VMs and containers imple-

mented through KVM and LXC technologies, respectively. The proposed layers in the framework base, application, and instance support the MEC system for migrating a service from a single MEC system level to another. The framework was tested using applications such as games, RAM simulations, video streaming, and face detection. The container-based model demonstrated peak performance. The authors, however, identified that the security risks are higher with container-based implementations compared to VMs where the connectivity is solid while migrating.

- *Blockchain for securing migration:* Wang et al. [137] suggest employing Blockchain for resolving trust issues among entities on different domains while migrating.

Summary: Migrating the services from one MEC-based eNB to another is a unique and required function in the MEC context. This phase can be identified as one of the weakest occurrences of edge computing in terms of security. Employing security is questionable for the migration channel due to the latency concerns in live migrations scenarios. As the channel itself conveys executable content, exposure could lead to the impregnation of malicious agents into the edge infrastructure. Therefore, a security framework is a requirement to exploit the latency and security relationship for maximizing efficiency. Further, Blockchain solutions can be employed for securing the states and credentials in the migration process.

The versatility to migrate service from one functioning infrastructure to another is a feature that improves the realization of MEC in the current heterogeneous IoT market, which solves the demand for ubiquitous connectivity to service access in a geographically dispersed context. This fact is quite imperative for emerging use cases of autonomous vehicles resembling applications where self-reliance on storage and processing capabilities within the on-device environment is highly challenging with budgetary constraints. Conversely, hosting the service in a cloud-native scenario is unreliable considering the latency-associated requirements of upcoming applications. On the contrary, maintaining the service continuity subject to mobility-based 5G guar-

antees, however, challenges the deployment of migrating schemes for operating seamlessly [137]. As stated in Section 3.2, there are obvious threats surrounding this process, and even a less-capable adversary could exploit the system through service-impeding attempts. A particular migration process might range from the traversal of a single executable file to a cloning of an entire serviceable platform. Thus, a proper scheduling mechanism that confiscates a recording scheme for state logs is a major requirement. In addition, suspensions during the migration process are inevitable due to connectivity failures, bandwidth restrictions, or intended service disruption attacks perpetrated by adversaries. The resumption of service and retrieval of stateful logs are entirely reliant on the scheduling mechanism. Moreover, scheduling mechanisms that embed migration protocols are critical for ensuring security and restoring the service in unintended intermittent circumstances.

The migration process is dependent on the virtualization technology (i.e., either containers or VMs [138]). Containers are lightweight in comparison to VMs that are best suited for resource-scarce environments. The reason for convenience is also the pitfall of containers on self-reliance in the perspective of migration, due to the requirement of OS libraries to operate on a migrated environment. VMs are capable of executing on any host due to the migration of the entire executable constructs onto the foreign vicinity. Thus, establishing security protocols on container-based deployments is restricted by the resource availability and compatibility of the local and migrating service platforms.

Preliminary Solutions: A security framework as presented in the literature [139] is required to protect the migration process, due to the complexity of the migration content and their states. A single mechanism is inadequate to cater to the security requirements. Furthermore, Blockchain [137] can be employed to secure the migration channels and to guarantee trust among entities involved with the process.

Research Problems:

A service migration that is rapid and secure is a requirement raised by the edge computing paradigms for maintaining service continuation in mobility circumstances. Since this is a novel area, there are doubts in the research context.

- *How to migrate the services reliably from one MEC edge to another?*
- *How to secure the service migration channel?*
- *How to exploit the relationship between security and latency in the context of the performance?*

3.3 MEC Privacy

In spite of the privacy enabling factors inherited by the MEC paradigm due to its proximate locality (improved location privacy; edge acting as the trusted, monitoring, privacy provisioning agent), certain aspects of MEC should be investigated to apply sufficient privacy mechanisms for users. This section discusses several factors where privacy issues can be identified within the MEC context.

3.3.1 Data Privacy

The confidentiality of user data that is either stored, processed, or in transmission is considered data privacy. Any mishandling of such private data is known as privacy leakage. Enabling Big Data applications, facilitated by high bandwidth and ultra-low latency enhancements of MEC, will generate massive amounts of personal data in the future. Applications such as healthcare, banking, and crowd-sourcing are brimful of such sensitive data [140]. Moreover, a certain dataset of a user is disseminated to server placements administered among various telecom operators due to the open service platform of MEC. This poses a confidentiality issue. In the current era, Artificial Intelligence (AI) based pattern recognition techniques are applied to user data by certain companies to identify trends and interests. As these activities outsource data without user consent, privacy is violated conspicuously. Thus, the preservation of subscriber privacy is a vital concern for launching the MEC paradigm.

3.3.2 Location Privacy

Though Location Based Services (LBSs) enable various applications for MEC subscribers, exposure of the geo-location endangers the financial, entertainment, professional, and secrecy aspects of human life: hijacking, blackmailing, or ransoming situations are possible [141]. The user location could be revealed to the subscribed service legitimately, either intentionally or unintentionally, as a pop-up service request for location sharing, that the subscriber is consenting without proper assimilation, without the awareness of the consequences. Moreover, secondary mobile channels broadcasting the wireless transmission, apart from the direct channel, are bound for monitoring by eavesdroppers for tracking location [142]. These factors contribute to the violation of location privacy.

3.3.3 Identity Privacy

Impending tactile Internet and IoT concepts are expanding the scope of the cyber-space. A method is needed to identify the billions of entities and people comprising it via interfacing UEs [140]. Any knowledge-based (username), possession-based (Random Number Generator [RNG]), inherence-based (bio-metric) cyber-address—or Physical Unclonable Function (PUF) in case of UEs—is adaptable for proving one's identity. Identity is the key to safeguarding private information in the cyber-space. An adversary capable of replicating a user identity could access the entire data cluster mapped to that identity. Attacks such as UE tampering, UE cloning, and masquerading, which commit identity theft, are examples of privacy violations in the access network. In the core network, a cyber-invader capable of exposing the identity credentials of the users or entities is presented with the opportunity to exploit the MEC system. Thus, preserving the identity of the users with expedient mechanisms is a principal requisite for MEC deployments.

3.3.4 Authorized and Curious Adversaries

An authorized entity can capture data that is not proprietary to itself with honest and curious intent. The extracted data could then be used for user profiling, location tracking, and disclosing credentials [37]. Thus, user privacy is violated regardless of the motive of the intruder. Such an initiative can be taken by the infrastructure or a third-party service provider that subscribes to an MES. Due to the openness of the edge ecosystem, constricting these legitimate practices is a conundrum. Every entity engaging with the ecosystem should be aware of its responsibility. Moreover, a mechanism should exist to detect any misbehavior perpetrated by an entity or a user.

3.3.5 Computational Offloading

Computational offloading is an exploitable feature of the MEC system that could be monitored by adversaries to determine the location information [143]. Tracking such intervening attacks is questionable due to the complexity of the offloading process and the non-accountability of the attack origination. Moreover, MEC servers inherit the features to track the usage patterns of the subscribers from the contextual information and channel status parameters. This creates a distinct concern for subscribers with respect to their private information [142]. Thus, computational offloading is a feature that risks the privacy of the MEC system.

3.3.6 Service Migration

Migrating services from one MEH to another at the same edge level or at another edge level is possibly dependent on the scope of the subscribed service and the mobility of the UE. Certain services, such as AR or autonomous vehicles, demand service migration in order to satisfy service requirements. Thus, a cyber-eavesdropper capable of reviewing the service statistics of the MEH entities is in a position to track the user from the migrating service patterns [141], leading to a violation of location and usage pattern privacy aspects. This

vulnerability exists with compromised MEC entities that could monitor service statistics of the VI platform.

3.4 Chapter Discussion

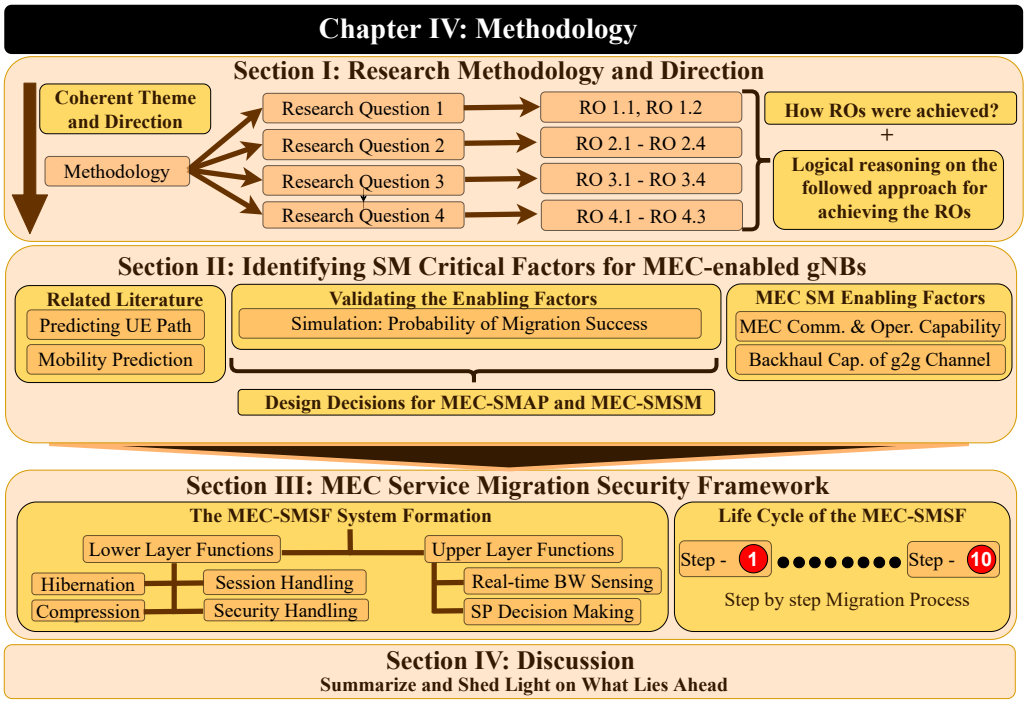
Despite privacy aspects of MEC being presented in Section 3.3, the primary research directive of this thesis doesn't cover privacy in its scope. This study has led me to understand that service migration is an aspect of edge computing that is worth further investigating. Though there is active ongoing research being conducted on the same topic, their focus on security and a way to manage the level of applied security and the cost it forms subjecting to latency is not an area that has been capitalized on. Thus, in this research, I have selected to focus on the security of service migration and to manage the applied security level, so that it would maintain service continuity.

CHAPTER

FOUR

METHODOLOGY

The methodology is an important aspect of this dissertation, as it states the scientific method that was followed to realize the validity of the proposed solutions. The investigations in Chapter 3 have led me to identify two main issues with the MEC service migration phenomena, (i.e. critical security concerns regarding the migration channel and the enforced latency appended on the communication overhead due to enhancing security) that formed the problem statement for this thesis. The solution for this problem cannot be simply formed by a singular construct but requires a holistic structure to develop the proposed solutions, implement them, and experiment on them to align with the scientific method to prove the presented hypotheses. This chapter introduces the said structure, MEC Service Migration Security Framework (MEC-SMSF), and its design proposed to solve the service migration-related security issues and to optimize the security and latency relationship in such a circumstance. Prior to presenting the MEC-SMSF, the methodology, and the preliminary investigations on the critical factors that enable service migrations in a pragmatic MEC deployment context are presented.



Chapter Organization

This chapter includes four sections. The intended coherent theme and direction expected for this research can be seen throughout this chapter. Section 4.1 describes the methodology in relation to the defined RQs both textually and in the illustrative context of Fig. 4.1 and Fig. 4.2. This section explicates how each corresponding RO was achieved and the logical reasoning behind each selection or decision made. Section 4.2 presents the essential findings related to investigations conducted to assimilate knowledge on critical parameters on the MEC edge platforms that involve service migration decisions. This section presented its own state-of-the-art analysis and validated its findings through simulations and emulations. Then, Section 4.3 presents the MEC-SMSF for the first time in this thesis, where its system formation in regards to its architecture is described and its operational life-cycle during a migration process is explicated in a step-wise sequence. Section 4.4 summarizes the content of this chapter and describes the relation of this chapter to the ensuing chapters of the thesis.

4.1 Research Methodology and Direction

The methodological approach for achieving the Research Objectives (ROs) specified in Section 1.3 is presented in this section. The specified ROs were defined to form a coherent direction for this Ph.D. study, where the research methodology was formed aiming at those milestones derived from the primary four Research Questions (RQs). This research is forthrightly mobile network related and focuses on the MEC emerging technology to solve its envisaged security issues while preserving the guaranteed service specifications, mainly the latency. Since this research is related to technologies that are quite novel, and their real-world deployments are a few years away, the experimentation, validation, verification, and proof of concept have to rely on simulations, and emulations conducted on prototype testbeds. And it is quite obvious to point out that extensive investigations had to be conducted to assimilate and envisage solutions to the selected problem. Since the state-of-the-art in regards to the network performance and security subjects are quite rich, the proposed solutions are integrating existing knowledge assimilated from literature, but in the context of the defined ROs and the RQs. Further, the Ph.D. problem is contriving diverse ROs due to its distinct domains of security, network performance, and emerging technologies. Thus, the defined ROs are completed and their contributions are published at various phases of this Ph.D. study; though, Table 1.1 lists the publications in accordance with the formed coherent direction.

Fig. 4.1 and Fig. 4.2 are describing the holistic research methodology of this Ph.D., where the corresponding RQs and their respective ROs are indicated to guide the reader on the selected research direction. The RQs of this Ph.D. was mainly targeting the 4 phases of investigation, identifying/ stating requirements for the solution design, development of the design, and testing and benchmarking. In Fig. 4.1 the methodology related to the RQs 1 and 2 is outlined. Under RQ1, a comprehensive investigation was conducted regarding the security vulnerabilities by specifying Threat Vectors (TVs) in a MEC deployment scenario and identifying probable attack vectors for each TV while proposing countermeasures from the literature for fulfilling RO 1. Out of

the identified TVs, architectural TV AR on service migration is a unique phenomenon to edge computing and MEC deployments, especially in an edge-to-edge migration scenario, where both security and network latency are prime research interests. Thus, this TV was selected as the critical and nuanced security threat of the MEC deployment, and the Ph.D. problem, RQs, and ROs were formulated accordingly.

In the context of service migrations in the MEC gNB-to-gNB (g2g) scenario, identifying the factors or parameters that are instrumental for service migration decision-making is imperative. MEC-based factors are reaching beyond typical mobile BS-based communication resources due to its computing and storage infrastructure. In fact, MEC operational capability, gNB communication capability, and backhaul capacity are the factors that were sought from the investigation. Since the actual real-world development of the MEC technology is still in the experimental stages, identifying the suitable technologies to launch the MEC serviceable platforms was the next milestone of this research. The adoption of lightweight and hypervisor virtualization technologies along with containerization is a viable approach to host dynamic serviceable instances with the flexibility to attach and detach from the infrastructure, as required by the service migration process, and specified in Section 2.2.

When investigating the security of the MEC service migration process, Man-in-the-Middle (MitM) threats are obvious. The most effective solution to circumvent the MitM threats is through a proper authentication mechanism prior to the service migration initiation. This is the best defense against the masquerading or impersonation threats perpetrated by resourceful adversaries. Thus, an authentication protocol prior to migration is proposed as a primary solution. Determining the impact of these threats eases the solution designs that follow in the next ROs. In fact, the impact is mapping the cause-and-effect of security threats, and leading the design, from where the security countermeasure should be devised from. Conversely, these security impacts are directed to formalize the security goals defined for the proposed protocol.

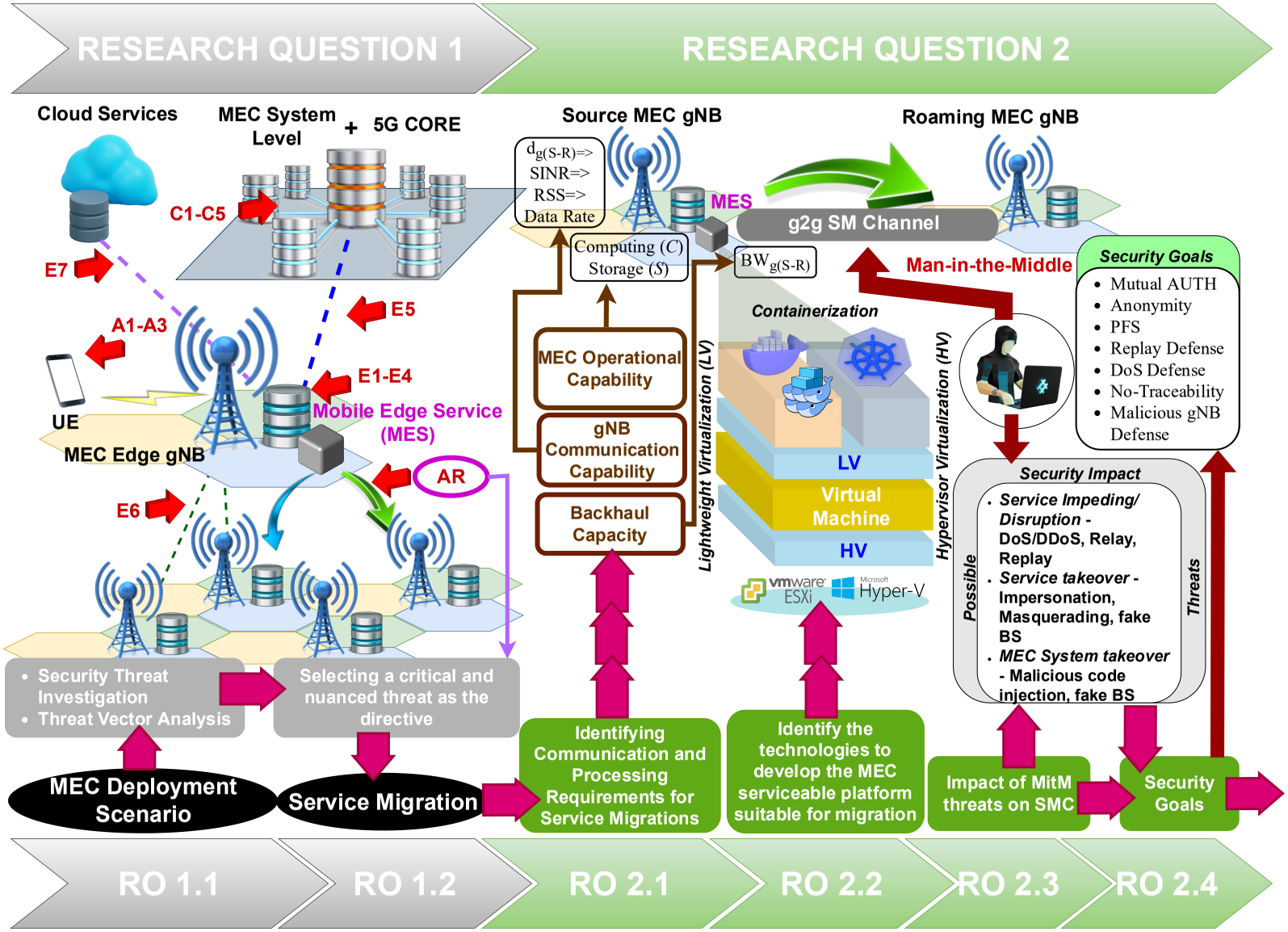


Figure 4.1: Research Methodology 1 of the Ph.D. Thesis

Fig. 4.2 depicts the methodology in relation to the RQs 3 and 4 respectively. RQ3 focuses on the development aspects of the Ph.D. solution, mainly the formation of the proposed security protocol, the development of the security framework for MEC migrations, and the security management function that would optimize the security-and-latency relationship of the migration process. From the specified security goals, under RO 2.4, integrative security mechanisms were identified to incorporate them into the security protocol and framework. The MEC Service Migration Authentication Protocol (MEC-SMAP) was then proposed adhering to the prescribed security goals and integrating the identified security mechanisms. In order to justify the feasibility of this proposed MEC-SMAP protocol, the actual development of the protocol was intrinsic. Thus, a prototype MEC platform was designed and developed with the virtualization technologies hosting the MES containers. And a TCP/IP-based communication channel was developed to establish the MEC-SMAP protocol between the Source gNB and Roaming gNB while constructing a content transfer protocol for enabling migrations. A Security Management function was formalized with dynamic security application capability enabled by the Security Profile concept, which is introduced under RO 3.4. In fact, security profiles specify different security levels, and an optimal security profile is selected by the proposed model dependent on the current residual bandwidth (i.e. ω_R) of the migration channel. This model is maintaining the migration delay or latency within the allowable limits for continual service operation.

The proposed MEC-SMAP is verified, formally (i.e. employing tools and logical analysis) and informally (i.e. proving the achievement of security goals) under RO 4.1. The feasibility of the MEC-SMAP was validated through both simulations and emulations described in Chapter 5. As the three major objectives of this Ph.D., MEC-SMAP, MEC Service Migration Security Management (MEC-SMSM), and the MEC Service Migration Security Framework (MEC-SMSF) that amalgamate all these solutions, were ready to be exploited at this stage. The MEC-SMSF framework was experimented on transferring a container with various distinct experimental scenarios.

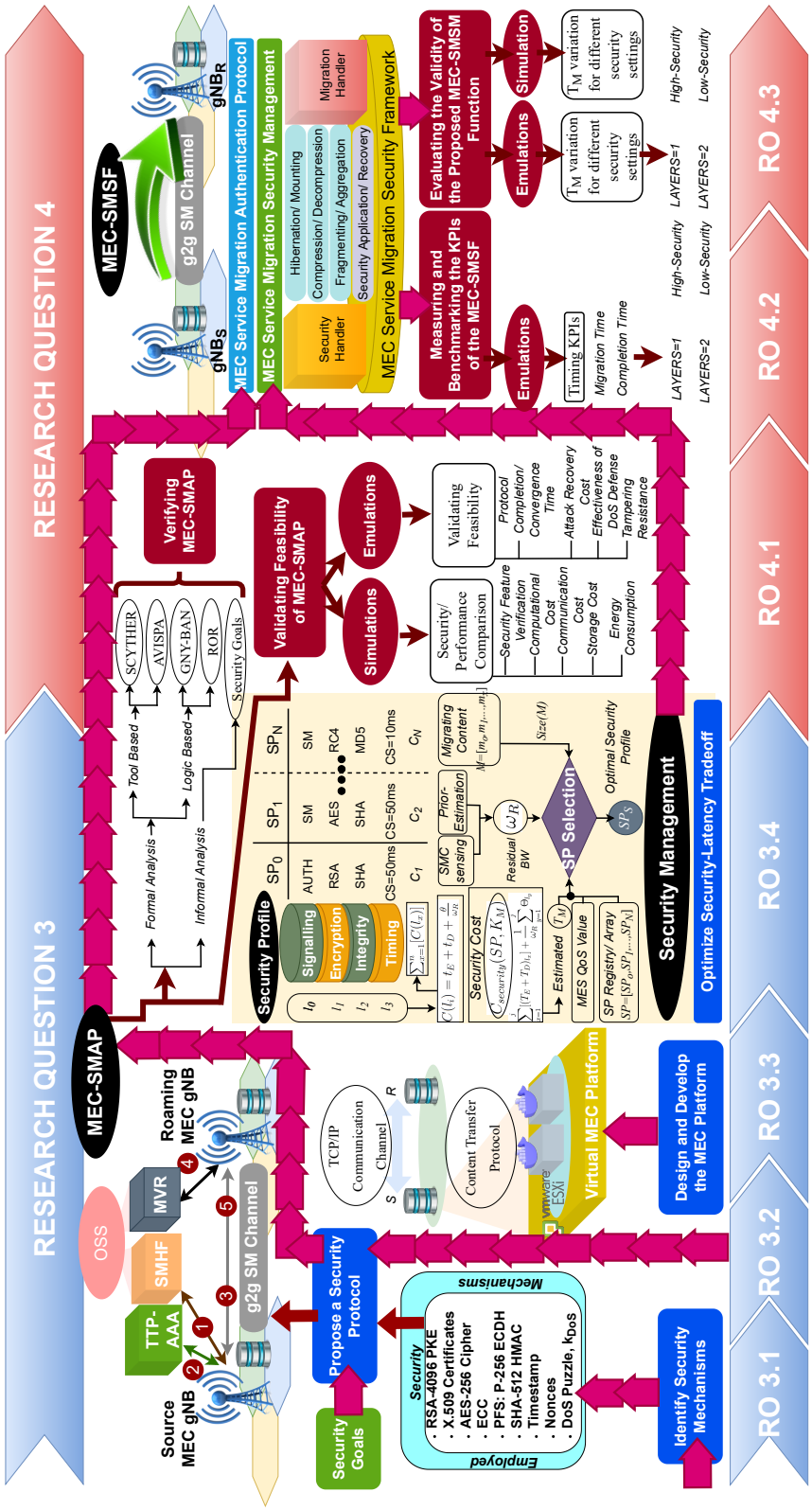


Figure 4.2: Research Methodology 2 of the Ph.D. Thesis

The timing values of encryption, decryption, compression, decompression, migration time, and completion time were considered the Key Performance Indicators (KPIs) of this MEC-SMSF framework. These KPIs were benchmarked to evaluate the effectiveness of the MEC-SMSM process. For RO 4.3, experiments in the form of simulations and emulations were carried out to validate the proposed MEC-SMSM functional model. The improvements that the security management function can impose on the migration delay under different security settings were evaluated to prove MEC-SMSM validity.

The next sections of this chapter are presenting the findings of the further investigations conducted to determine the requirements for MEC service migrations, and MEC development technologies; while presenting the proposed MEC-SMSF design.

4.2 Identifying Service Migration Critical Factors for MEC-enabled gNBs

In accordance with the methodology of this Ph.D. study, it is important to identify and understand the factors that can impact service migration decisions beforehand. In spite of their significance for service migration decision-making (i.e. when and where to migrate?), understanding them is the key to managing the security levels beneficial for the formation of the strategies later on. This section will introduce the identified factors and will validate them through simulations.

4.2.1 Related Literature

For service migration processes to run seamlessly, a considerable level of intelligence is required at the MEC edge (i.e. preferably by the MEPM) equipped with situational awareness. Prior assimilation on the dynamics of the UE or even the knowledge of where it might travel in the next instance would be beneficial for service migration decision-making. In fact, prediction can be a way to enhance the effectiveness of the migration process that introduces a new

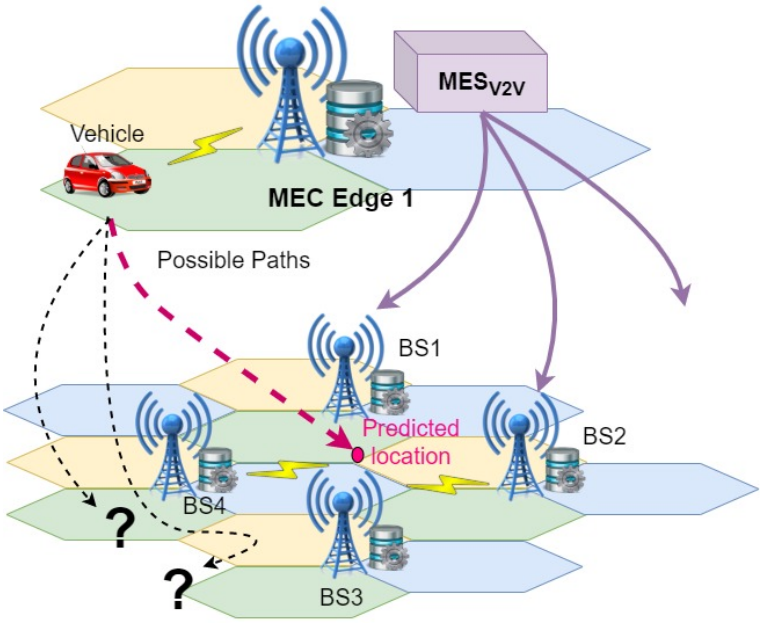


Figure 4.3: Predicting the UE/ AV Path in Mobile Networks

dimension to excel the designs. Thus, the prediction of the UE path aids to determine the mobility plan for the migration process. Through this process, intended critical factors can be identified.

4.2.1.1 Predicting UE Path in Mobile Networks

Predicting the User Equipment (UE) path, or the subsequent coverage area (i.e. gNB coverage or a cell contrived by a sector antenna of a gNB) is important for MNOs for various reasons other than enhancing service migration. Prediction aids to evolve current dynamic mobile traffic management models, and business models by pre-loading services to cater to user preferences. Though the ideology is simpler in retrospect, predictions based on mobile network statistics are inconclusive due to their rapid mobility and associated interference as illustrated in Fig. 4.3.

Nadembega et al. in [144] propose a mobility prediction scheme considering probability and Dampster-Shafer processes for mobile networks. The intention is to allocate radio resources to the predicted trajectory of the user for optimizing the limited resources with a desirable level of QoS. A Destination

and a Mobility Path Prediction Model (DAMP) is formed with input parameters extracted from user knowledge, and regular spatial and temporal patterns.

4.2.1.2 Mobility Prediction in Service Migration

The [145] exploits the trade-off between the traffic overhead and QoE in service migration scenarios with Follow Me Cloud (FMC) deployments. In this approach, the sequence of transferring data is planned according to the mobility pattern and estimated load of the data centers. This solution includes a reception throughput estimation scheme, a hand-off time estimation scheme, and a migration management scheme for the considered service. The throughput was estimated based on a matrix formed with values estimated through a semi-Markov process. A similar strategy was employed for estimating the hand-off time with the distribution of probabilistic values. The migration management process is selecting which service to migrate at a specific time interval determined through the maximized throughput and minimal hand-off time computed with algebraic means. Though, the resource utilization at the FMC infrastructure was not taken into account when performing the selection.

4.2.2 MEC-based Critical Enabling Factors for Service Migration Processes

The factors or parameters enabling the governance of the service migration phenomenon can be specified from the communication and operational capability of the edge node, and the backhaul capacity of the migration channel.

4.2.2.1 Communication and Operational Capability of the MEC Edge Node

Communication Capability : Signal-to-Interference-Noise-Ratio (SINR) measurement of a radio channel gives a good perception of the Received Signal Strength (RSS), which corresponds to the communication capacity of the migrating MEC node. The aerial distances derived from geo-locations (assuming obstacle-less line-of-sight radio links) can be employed to determine

the SINR, where data rates in bps can be computed. A permissible communication range (R - minimum data rate, SINR level [146], or a number of edge nodes) should be established to conduct the selection process for the eligible roaming gNB.

Operational Capability : the operational capability of the considered MEC edge node can be determined by the consuming computing (C) and storage (S) capacities at the considered instance. The factors of latency, jitter, and priority level of the MES are contributing to the C and S parameters. QoS Class Identifier (QCI) standards and specifications are aiding to model these two factors in line with the emerging applications and services [147].

4.2.2.2 Backhaul Capacity of the Migrating Channel

The edge-to-edge Bandwidth (BW) of the backhaul link is critical for modeling the service migration process. The existing backhaul links are typically employed for signaling purposes with limited capacity. Though, novel services tend to utilize these links for improving QoE aspects with embedded intelligence. As migrations are less-occurring events, there is no guarantee that available backhaul capacity would be sufficient for migration initiation. For the derived validating model, migration time (computed from available link capacity and size of the migrating content) for a specific MES can be considered a viable input parameter.

4.2.3 Validating the Proposed Enabling Factors

The evaluation was carried out as a simulation to determine the probability of success in migrating a MEC application serving a UE, to a MEC-enabled gNB where the UE is predicted to be traversing. Two scenarios can be considered to validate the proposed factors. The two scenarios are:

Legacy Scenario : The optimum gNB for initiating the migration was selected based on the serving communication link capacity (i.e. data rate) as presented in [146].

Proposed Scenario : The optimum gNB was selected based on the resource capacity (i.e. computing and storage) of the corresponding MEC infrastructure in addition to the communication link capacity.

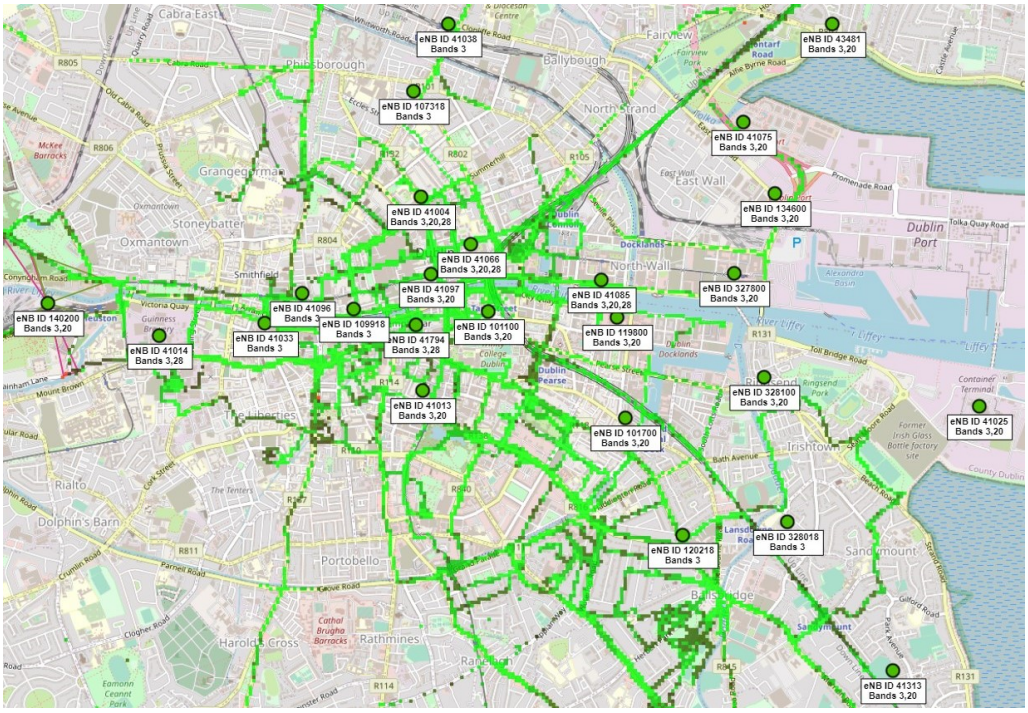


Figure 4.4: The Site Map with 25 gNBs in Dublin

In order to conduct the simulation, 25 gNBs located at Dublin City, Ireland (i.e. longitude $53.3243282° \sim 53.3654212°$, and latitude $-6.2956804° \sim -6.2071241°$) was considered, as shown in Fig. 4.4, with their actual geolocations and bandwidths extracted from [148]. All the gNBs in this map were assumed to be enabled with MEC capability, while the coverage area is dense with mobile traffic due to their urbanized localization. Further, MESs were incepted based on the QCI levels 1,2,3,4,5,6,7,8,9,70,80,84 to randomize the C and S consumption. The 3GPP propagation model in [149] was followed to compute the data rates. In addition, parameters in Table 4.1 were considered to perform the simulation, where R was considered as 10.

The simulation was carried out for continuous 500 trials where the UE location was randomized within the grid to determine the success of launching the migrated services at the roamed MEC edge node based on its resource

Table 4.1: General Simulation Parameters

Parameter	Values
Number of gNBs	25
Average gNB BW	15 MHz
Maximum transmission power of a gNB	46 dBm
Average resultant antenna gain at gNBs	5 dBm
Average resultant noise power	-92 dBm
Number of MESs Normalized μ / σ	100/ 90
Maximum gNB computing capacity	100 GHz [150]
Maximum gNB storage capacity	100 TB [150]
Maximum computing capacity of a MEC App	60 MHz
Maximum storage capacity of a MEC App	100 GB

availability. The mobility model of the UE considered for this simulation was determined based on randomized locations within the grid as stated. Realistically, the localization of the UE would be set within the daily mobility/commuting paths of the UEs. However, such a viable data set could not be obtained to perform the simulation. Note that the specific mobility model used does not affect the results of the simulation, as it is contrasting the selection of the MEC gNBs, considering two scenarios; the mobility model will be common for both scenarios. It is observed from Fig. 4.5, the success rates of both legacy and proposed scenarios are 33.4% and 98.8% respectively.

Another simulation was conducted to determine the impact of R for the two considered scenarios. It is obvious that with more options at hand, the probability of success becomes improved with the increasing R for the proposed scenario. The dynamics of the optimum gNB selection is illustrated in Fig. 4.6. Here, Legacy scenario 2 represents the selection process conducted only via the MEC resource capacity. Though, increasing the permissible range would lead to selecting gNBs with lesser communication capability (i.e. data rate). Therefore, a minimum data rate requirement should be specified in accordance with the MES QCI value.

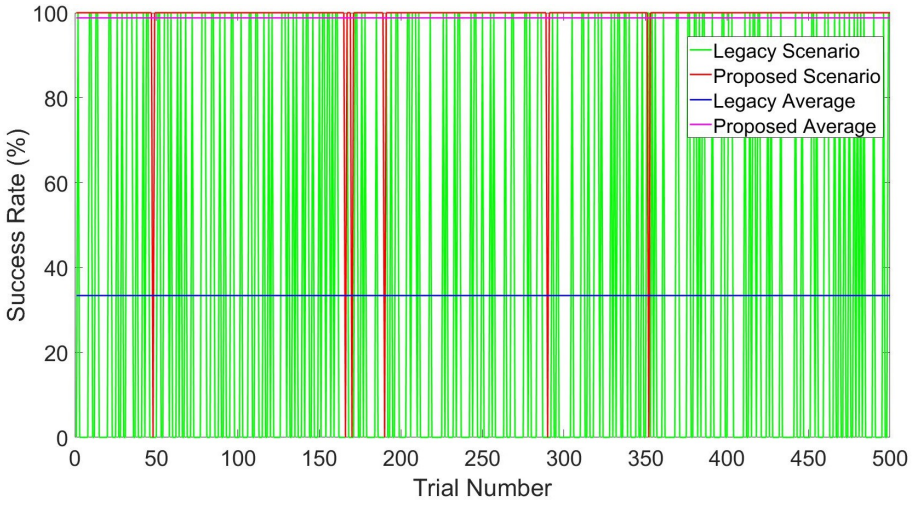


Figure 4.5: Simulation results on the success of launching the migrated services

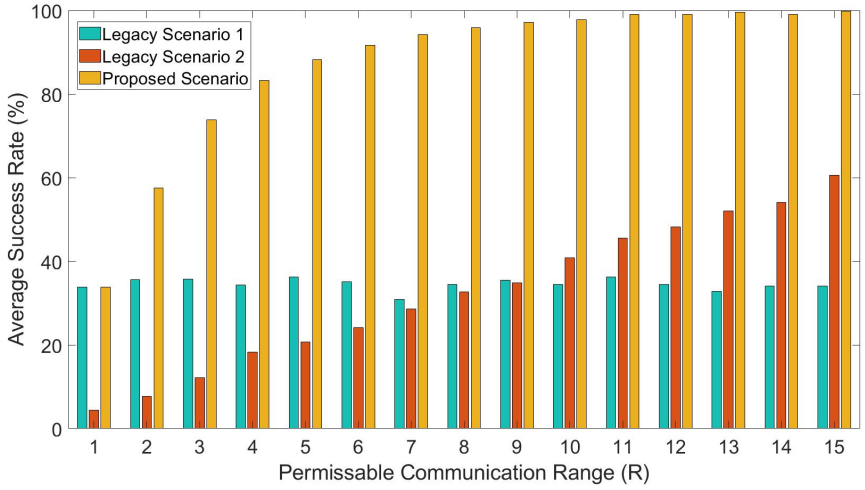


Figure 4.6: Simulation results on the success of launching the migrated services

4.2.4 Concluding Remarks of this Study

The presented validation proves the effectiveness of the identified parameters defined in terms of communication, the operational capability of edge nodes, and the backhaul capacity of the migrating channels. These factors will provide a baseline for formulating solutions to current issues of service migration: security, latency, mobility, and handover management.

4.3 MEC Service Migration Security Framework

In this section, the proposed security framework established for service migration scenarios of MEC, its purpose, and its associated features are emphasized. MEC-hosted services are envisaged to be operated as virtualized entities that can be decoupled from their underlying infrastructure or firmware to enable rapid and dynamic migration. To this end, for such entities to become more suited for migration, services can be hosted in lightweight containerized entities for ensuring the least impact on the performance metrics of the virtual platforms [151, 15]. More subjectively, Mobile Edge Services (MESs) can be launched as containers (i.e. docker, or Linux) while its parent platforms Mobile Edge Hosts (MEHs) can be launched as Virtual Machines (VMs) as specified in [16, 152], and already described under Section 2.2, and validated under Section 7.3. Migration of a particular MES is represented by transferring the contents of a container, or a hibernated image of the said container. In such an instance, the hibernated image of the MES should be re-configured and launched at the roamed gNB MEC environment after the migration.

4.3.1 The MEC-SMSF System Formation

The framework illustrated in Fig. 4.7 is proposed to perform the task of service migration within the g2g channel, embedded with optimal security strategies that guarantee maximal service level efficiency. Thus, this framework is termed as MEC Service Migration Security Framework (MEC-SMSF) throughout this thesis. The main function of MEC-SMSF is to transfer or migrate the MES service instance from the gNB_S (i.e. gNB where the MES is currently operating) to gNB_R (i.e. gNB where the MES is intended to migrate and relaunch). The migration-related functions of the MEC-SMSF are explicated below.

4.3.1.1 Lower Layer Functions

These are the functions handled by the environmental constructs available within the MEH hosting the migrating MES. In fact, these functions are directly

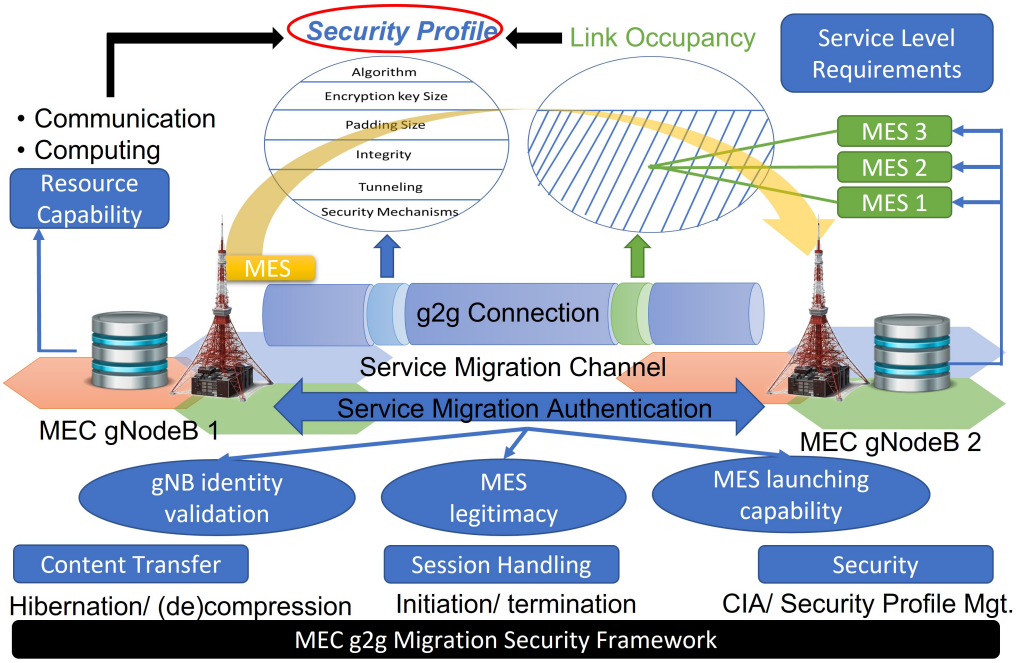


Figure 4.7: MEC Service Migration Security Framework

involved with the migration process. Further, the virtualization infrastructure within the MEH is acting with the data transfer interfaces of the MEH to make this process successful.

4.3.1.1.1 Hibernation This function refers to the temporary shut-down of the operating MES to prep the container for migration once the migration interruption is invoked. However, this operation could take different approaches depending on the circumstance and the allowance for the service downtime from the service requirements. With Docker, there are several constructs that can achieve this intention. The commands of `export`, `save`, and `commit` are such approaches that can achieve this function depending on their merit and situational awareness [153, 154]. As presented by [46], docker approaches are viable for variant migration strategies and are suited for V2X services.

4.3.1.1.2 Compression This function is quite important in the context of migration, as well as for security. According to the experiments conducted (i.e.

in Section 6.4.2), compression can achieve up to 37% reduction in the original migration content size. This reduction is quite vital in the context of bandwidth utilization of the service migration channel and adds a confusing aspect to the migrating content.

4.3.1.1.3 Session Handling Similar to any data transfer process, session handling is an imminent function in the transport layer, whether the transmission mode is Transmission Control Protocol (TCP) or User Datagram Protocol (UDP). The protocol to be followed can be selected depending on the traffic classification required by the QCI of the MES application. A session handler launched within the MEH that is hosting the migrating MES is capable of performing this task, where maintaining the connection with the gNB_R channel interface and ensuring the reliability of the transmission is its key function.

4.3.1.1.4 Security Handling Ensuring the Confidentiality, Integrity, Availability, Accountability, and Authenticity of the migrating content is a prime requirement of this framework. Due to its significance and complexity, it can hold, this function was designed to be performed by a separate security handler operating within the same MEH. Obviously, this handler will be in charge of applying the relevant security mechanisms to the migrating content, as zero security is not an option in this context. In addition, the security management function explicated in Section 6.2 is operated by the same handler to optimize the usage of security profiles.

4.3.1.2 Upper Layer Functions

As indicated in Fig. 4.7, monitoring the resource capability of the MEC and the hosting gNB environment is essential to uplift the intelligence and situational awareness required by the MEC-SMSF. The factors identified in Section 4.2 are quite vital for attaining this feat.

4.3.1.2.1 Real-time Bandwidth Sensing The overall g2g channel utilized for the service migration process should be monitored in real-time with a μs or

ns resolution, as this utilization factor is important to decide the security level (explained further in the Section 6.2).

4.3.1.2.2 Security Profile/Setting Decision Making It is shown in Section 4.2 that the factors of communication capability and operational capability of the MEC edge of the gNB_R, the backhaul capacity of the service migration channel, and the service level requirements of the migrating MES are aiding the decision to select which MEC enabled gNB should the MES should be migrated to. After this selection, however, during the migration process, these factors could still change due to various reasons as these resource-based measures are dependent on the other operating MESs. Thus, the security level of the migration process or the security profile should be decided upon in case such a change has taken place with the resource factors. This decision-making is vital to perform the service migrations seamlessly in dense traffic situations.

In addition, MEC-SMSF is formed for two primary purposes other than the migration initiative. Those purposes are:

1. Mutual authentication of entities engaged in a service migration scenario
2. Optimal security management to maintain service level guarantees

As these two purposes are the main contribution of this Ph.D. thesis, and they are being thoroughly investigated, solutions are designed, developed, and validated in the upcoming chapters of this thesis.

4.3.2 Life-cycle of the MEC-SMSF

Fig. 4.8 illustrates the typical operation of the proposed MEC-SMSF framework. Each stage of this process is described below. The initiation of this process is assumed to be taken place after the authentication process and the migration roaming agent selection processes.

(1) - The Geo-tracing of the UE with the specified prediction capabilities in Section 4.2 would notify whether this UE is opting for a service migration or not. In such a scenario where migration is opted for by the system, MEPM

will invoke the service migration function through a Migration Handler(MH). This MH will invoke a Security Handler (SH). Once invoked SH will extract the agreed upon security profile list (i.e. end of the security protocol) from the security profile registry.

❷ - The MH sends an interrupt to the MES indicating to halt the service.

❸ - Once halted, the MES will hibernate its container to an image file.

❹ - The MES image will be compressed.

❺ - The content of the compressed image will be fragmented for convenience in migration. The size of the fragment is dependent on the TCP/UDP window, BW of the channel, or the overhead/digest of the highest security level.

❻ - SH will apply the corresponding security profile to the fragments sequentially and conveys them to the gNB_R through the g2g channel.

❼ - Until all the content is transmitted, the channel state and BW utilization will be monitored by the SH. If there are any significant changes in the BW utilization, the security profile will be switched accordingly to manage the security level.

❽ - All the migrated fragments are aggregated into a temporary registry by the roaming SH. After the migration is complete, the aggregated content is decrypted to obtain the compressed image.

❾ - Decompression of the compressed image to form the MES image.

❿ - The formed MES image is mounted to the roaming dockerized environment and the service is configured and initiated.

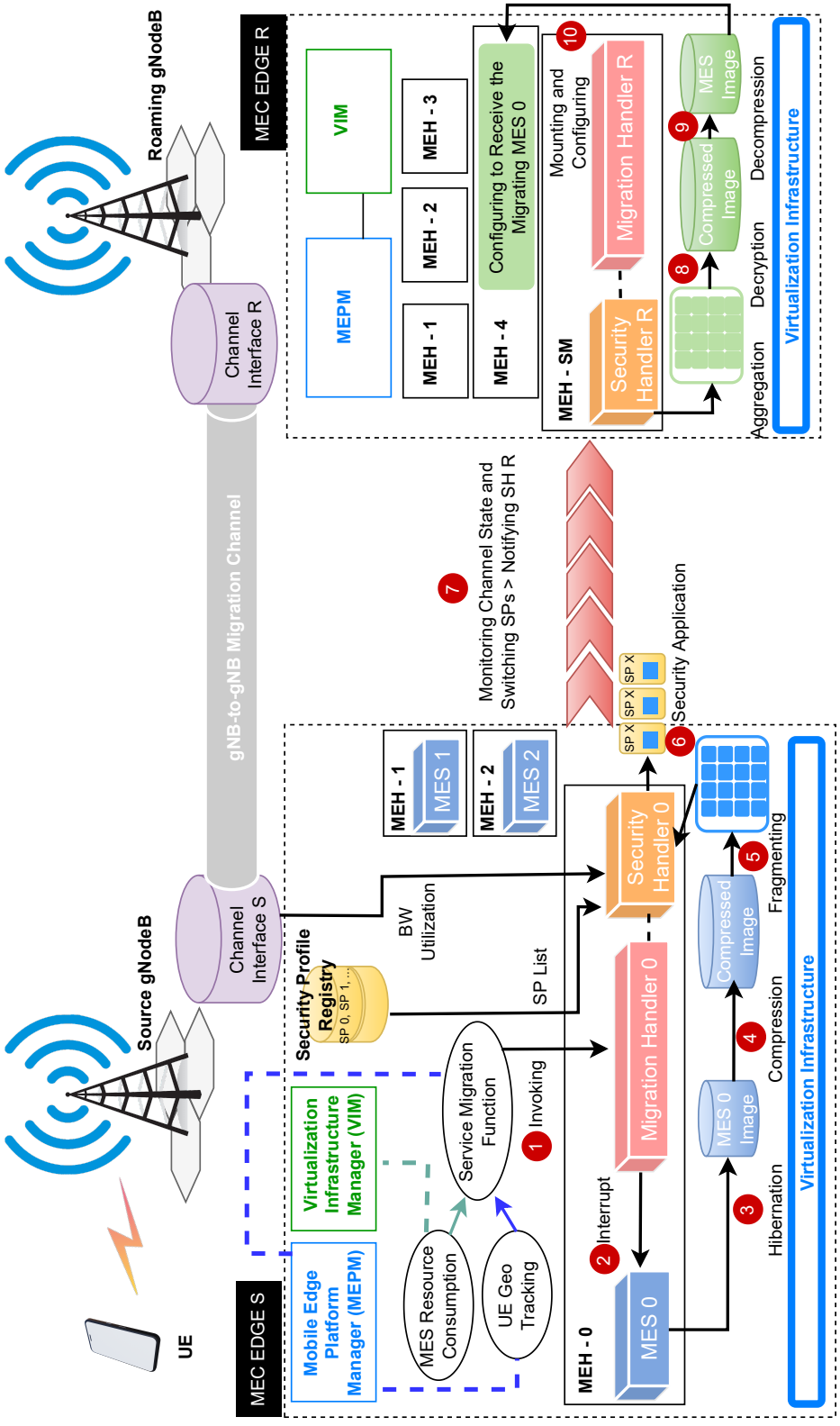


Figure 4.8: Life Cycle of MEC Service Migration Security Framework

4.4 Discussion

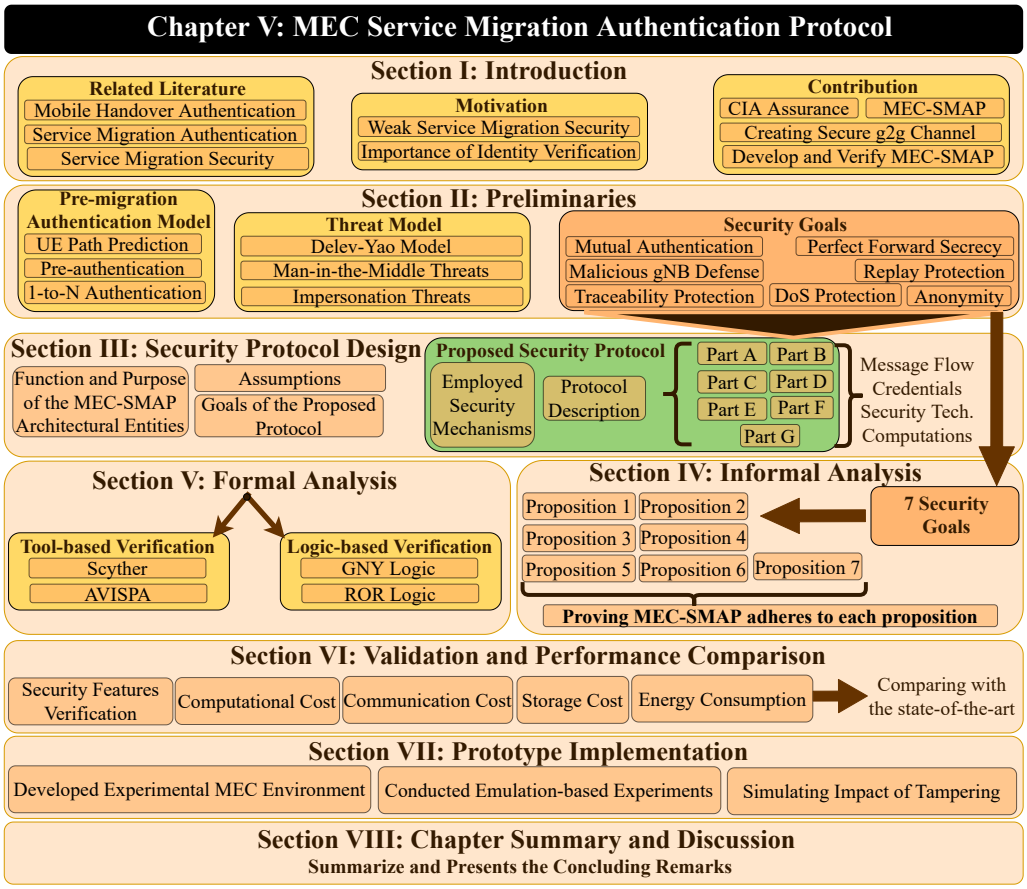
In this chapter, the research direction and the methodology followed for this thesis was presented. The research direction of the Ph.D. is specified in the Section. Accordingly, a study conducted to identify the resource-based factors imperative for service migrations was presented. Its validity was proven with simulation and emulation-based experiments. As the core contribution of the Ph.D., and the primary solution for this Ph.D. problem, MEC-SMSF was introduced. The intended functions of its design and the migration life-cycle were presented. The presented content paves the path to the upcoming chapters, where the scientific and technical contribution of this thesis is presented.

CHAPTER

FIVE

MEC SERVICE MIGRATION AUTHENTICATION PROTOCOL

As specified in ETSI MEC standardization, edge computing serviceable infrastructures are running on virtualization technologies to provide dynamic and flexible service instances to cater to User Equipment (UE) of various formations to accomplish diverse use cases. Since the inception and operation of the services are executing at the edge-level gNodeBs (gNBs), migration of services between gNBs is an imminent occurrence in edge computing that is contriving challenges to its feasible deployment. Security and service level latency requirements are vital parameters for such service migration operations conducted through gNB to gNB (g2g) connecting channels. In this chapter, the focus is to ensure identity verification among the parties involved in a service migration through authentication and to secure the migrating content through a robust g2g channel establishment. The proposed authentication protocol was designed in accordance with the MEC architectural standardization. The proposed protocol was verified employing four different formal verification techniques: Scyther and AVISPA verification tools; GNY and ROR logical approaches. Further, the proposed protocol was developed in a test-bed environment emulating the MEC system, with an integrated 5G Core network.



Chapter Organization

As this is one of the main contributions of the thesis, this chapter consists of 8 sections. In the introductory Section 5.1, the motivation of this research is presented along with the state-of-the-art, while corresponding contributing aspects of this chapter are mentioned. Section 5.2 introduces the pre-migration authentication model followed in the MEC-SMSF, while the considered threat model and the security goals of the proposed protocol are specified afterward. In Section 5.3, the design aspects of the security protocol are discussed and aligned to the considered service-migrating MEC system model. Here, considered assumptions, and overall goals of the service migration protocol are presented while the different parts of the protocol are described extensively.

The informal analysis of the proposed protocol is presented in Section 5.4, while the formal analysis results employing Scyther, AVISPA tools, and GNY, ROR logics are presented in Section 5.5. Section 5.6 presents the respective computational, communication, and storage cost of the proposed protocol along with embedded security features, and contrast them against the current literature. The details of the developed prototype MEC environment are presented in Section 5.7 and discusses the results of the experiments conducted regarding the effectiveness of the employed security mechanisms. Section 5.8 summarizes the content of the chapter and discusses the effectiveness of the MEC-SMAP design.

5.1 Introduction

5.1.1 Motivation

The emergence of edge computing paradigms has introduced the concept of service migration to cater to the heterogeneous IoT device's ubiquitous connectivity over the mobile network. The MEC-based services are offered from the nearest MEC-enabled gNB to the subscriber. Since the service instance or the program executing at the edge platform is originating there, such a service instance is not available in other MEC gNBs. In a situation where the subscriber is traversing beyond the range of the currently serving MEC gNB, the service instance should be migrated to a gNB with MEC capabilities that is in the proximity of the subscriber-roamed location. Once migrated and configured to the roamed MEC infrastructure, offered service to the consumer continues through the communication channels of the roamed gNB. The Quality of Service (QoS) and Quality of Experience (QoE) aspects of the offered MEC-based service are entirely dependent on the seamless operation of the migration process. The latency or a delay caused in the migration process will result in disruption of the service to the consumer device; thereby impacting both QoS and QoE factors negatively. Thus, service migration in edge computing platforms is a weaker aspect of MEC that forecasts inevitable issues.

In a typical BS-to-BS communication, specialized authentication is not required as all BSs are registered under an MNO. Though with the advent of 5G, local operators are granted the ability to launch services in the mobile network, and such operators are not quite trustable due to the scalability of 5G. There is always a possibility of a fake gNB being launched by an adversary with replicated communication protocols. Since service migrations are becoming a frequent occurrence in 5G, gNB-to-gNB (g2g) communication is becoming a regular function for emerging networks. In addition to the impact it causes on the User Equipment (UE) communication, implications to the service migration process would be severe. This severity is due to the fact that service migration is conveying mostly executable content or sensitive credentials between the gNBs. The sensitive information in the migrating MESs is prompting privacy issues in addition to security concerns. Mishandling or attempting to mishandle the user information during the migration itself is compromising the privacy of the user, and anonymization is merely a passive defense. Thus, validating the identity of the gNB prior to migration is critical for 5G consumers to ensure both security and privacy. An authentication mechanism can validate the identity of the 5G gNB, and establish a secure migration channel afterward. Though, a Trusted Third Party (TTP) should be engaged as the identity verification (mutual authentication) of both entities/ parties cannot be pursued in an ad-hoc or peer-to-peer manner.

One might think that a typical authentication mechanism as presented in [155] or [156] could be sufficient for this authentication mechanism. Though, service migration is a unique process, and requires a specialized protocol to perform the eligibility and verification (resource availability of the roaming gNB, network capacity availability of the Service Migration Channel (SMC), and Validation of the migrating virtual instances) mechanisms embedded in it. Further, different security profiles (i.e. a standardized template of different security mechanisms that states the sequence and the purpose of those mechanisms) are ought to be applied to the SMC, depending on the application specifications. Thus, the selection of the SMC security profile should also be communicated via this protocol establishment. Hence, these aspects are uniquely addressed in this protocol. The details of the formation of the security

profile are further elaborated in Section 6.2.2.

5.1.2 Related Literature

Zhang et al. in [155] propose a handover authentication protocol for 5G-based Heterogeneous Networks (HetNets) called RUSH. As rapid handovers are imminent with 5G HetNets, mutual authentication and key agreement become imperative requisites to ensure secure identity management with emerging applications that feature high dynamism. The authors have further employed chameleon hash functions considering their trapdoor collision property, and blockchain for its tamper resistance. RUSH has been verified with formal logic and model-based methods while its computation and communication efficiencies are leading among the existing schemes in contrast. Though, the proposed RUSH scheme is not directly related to service migrations, the context in which it forms a secure channel between 5G HetNet BSs or access points is quite relevant for this study.

Zhang et al. in [151] propose a blockchain-based secure edge service migration framework called Falcon, that enables VMs or containers to be migrated as mobile agent-based carriers to make the migration process more flexible. This framework employs a selection algorithm to maximize the migration benefits in line with the service quality. The immutable alliance chain decentralized to edge clouds is improving the performance of the Falcon with blockchain inclusion. The identity verification and management engaged in migration is a lacking aspect of this protocol, where a comprehensive security solution is required apart from blockchain. Cui et al. in [157] introduce a fountain codes-based jamming strategy for service migration scenarios of edge computing environments. This solution contrives a set of Relay nodes to conduct cooperative jamming, which would eventually mislead and deteriorate the illegal eavesdropping quality of the migration channel. This strategy, however, does not provide a solution for trust/ authenticity verification among migration entities.

An authentication protocol for service migration scenarios in cloud computing was proposed by Karthick et al. in [156]. This protocol was targeting

vehicular applications that require migrations between two clouds, and a registration entity is performing the communication of resource allocation securely. Though this protocol is been validated with AVISPA, there are evident issues such as needless signature exposure that opt for reuse threats, ill consideration of perfect forward secrecy, and Denial of Service (DoS) threats. In addition, the lack of a development environment, or any simulated performance metrics indeterminate the feasibility of this protocol.

Braeken et al. in [158] present an edge-based secure architecture for healthcare applications to perform secure key management. The proposed protocol is targeting Mobile Augmented Reality (MAR) based applications, where Man-in-the-Middle (MitM) type, Distributed Denial of Service (DDoS), impersonation, malware injection, and Side Channel Attacks (SCA) attacks are in its scope of the defense. The credentials employed from the edge perspective along with the DoS mitigation approaches are utilized for this research attempt.

5.1.3 Contribution

To the best of our knowledge, there isn't any literature available addressing the security issues of edge-to-edge service migration scenarios. Our work can be considered as the pioneer attempt on solving the security concerns of service migrations. In this chapter, A holistic security protocol is proposed for authenticating and establishing a secure channel for pursuing service migrations. The main contributions of this research are stated below.

- Proposing a communication protocol for service migration instigation from a MEC architectural standpoint.
- Ensuring the authenticity and integrity of parties engaged in the service migration process through a federated identity verification approach.
- Securely conveying the credentials, or parameters that enable the formation of a security profile.
- Establishing a secure g2g channel for service migration content transfer.

- Validating our claims through formal analysis employing the Scyther tool, AVISPA tool, Gong, Needham, and Yahalom (GNY) logic, and Real-Or-Random (ROR) logic to show the robustness of the proposed protocols.
- Conducting an informal analysis to verify compliance with the proposed security goals.
- Developing a prototype MEC environment to deploy the proposed protocol in line with the requirements of the framework.

5.2 Preliminaries

5.2.1 MEC Pre-Migration Authentication Model

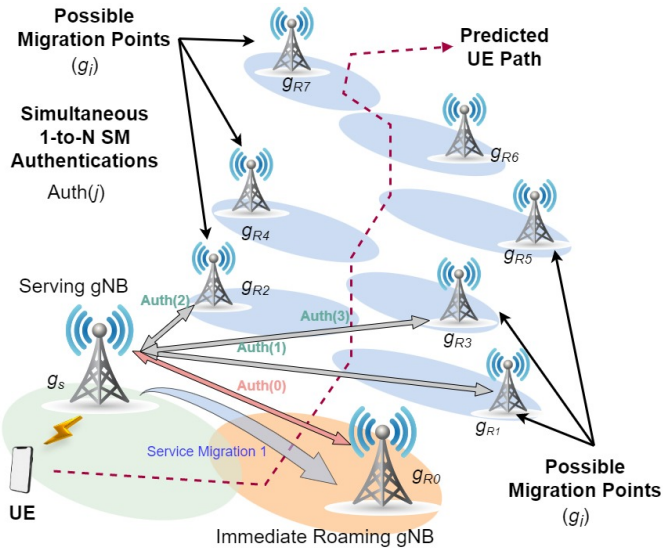


Figure 5.1: Proposed MEC Secure Service Migration Model

In order to improve the efficiency in service migration scenarios, UE path prediction models can be considered as specified in [145, 144]. For this prediction, the prior knowledge on the UE mobility/velocity (i.e. traced via GPS coordinates on a timely basis), date, time, and condition of the weather are intrinsic. With such path prediction models, it is possible to determine the predicted GPS coordinates of the UE. Thus, simultaneous 1-to-N authentication

can be conducted among the gNBs that are positioned on the predicted path of the UE prior to the migration initiation, thus, pre-migration. Since this authentication process is conducted preemptively, all possible gNBs within the immediate coverage domain with MEC-capability are considered potential authentication parties. As illustrated in Fig. 5.1, possible migration points are notified as g_{Ri}, where each authentication session is specified as $Auth(i)$. Though this process is simultaneous, the priority is given to the proximate g_{Ri} points. The selection of the i factor or the N value is a matter to be explored subjective to the UE perspective (i.e. UE trajectory and mobility), Mobile network perspective (i.e. MEC-enabled gNB density, individual gNB coverage), and the Application specific perspective (i.e. application QoS Class Identifier/QCI, allowable downtime for service migrations, application availability, and scalability). In this thesis, however, the focus is on establishing a single $Auth(i)$ session with the intention of maximizing security for that specific session.

5.2.2 Threat Model

To examine the robustness of the proposed protocol, we employ the Delev-Yao (DY) [159] threat model. The adversary's capabilities are as follows

1. The adversary (A) has total control over a wireless channel where an attacker may remove, modify, or inject legitimate messages.
2. A can only guess one credential in polynomial time because simultaneous guessing of values as identification or password is impossible.
3. A can intercept many messages to launch a traceability attack.
4. A can act as a middleman and launch a Man-in-the-Middle (MitM) attack. Adversary stealthily relays/possibly modifies communications between two parties that believe they are conversing directly with each other.
5. A may also reveal some session secrets.
6. A can also obtain the private keys of communicating parties.

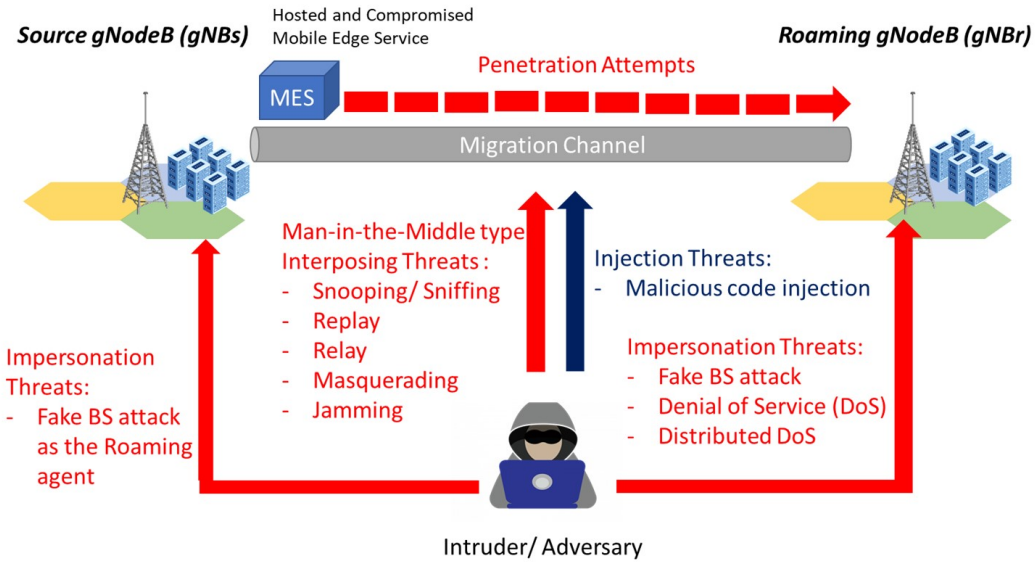


Figure 5.2: Considered Threat Model

Though the DY model was selected as the standard model for this security analysis, Fig. 5.2 indicates the formalized non-standard threat model along with targeted attacks/ threats considered for this protocol. As indicated in Section 3.2, the most vulnerable or critical component of the service migration TV is its channel shared among the two gNBs involved in the migration. For this channel, any MitM-type threat (i.e., snooping, sniffing, Replay, Relay, Masquerading, or jamming) or injecting threat (i.e., malicious code/worm injection) can be perpetrated as penetration attempts. The consequences of injection or instilling attempts are more severe considering the nature of the conveyed content being executable, and detecting such compromised content is arduous after the conclusion of the migration procedures. In addition, impersonation threats (fake base station, DoS, or DDoS) can occur targeting either the source or the roaming agent. It is understood that threats of DoS and DDoS can affect the MEC system more negatively, considering the latency-prone nature of the hosted emerging services.

5.2.3 Security Goals of the Proposed protocol

The following are the security goals [160, 161, 155] that the designed authentication technique must meet.

- Mutual authentication: It states that before sharing any private or personal information, communication parties must check each other's legitimacy.
- Anonymity: The identities of communication parties should not be communicated in plain text over insecure public channels.
- Perfect Forward Secrecy (PFS): This concept ensures that even if an attacker is capable of acquiring long-term credentials, revealing the prior session keys or secret content within the messages is not viable.
- Replay attack protection: It ensures that it is impossible for an attacker to replay the old message.
- Protection from Denial of Service (DoS) attack: It is difficult for an attacker to create network congestion by sending reused messages. In this goal, the scope of the DoS threats is limited to request-based DoS threats at the application layer directed towards the MEC gNB interfaces. Such attempts can occur in a distributed fashion through botnets. However, frequency jamming-based threats were not considered as they are outside the scope of this thesis.
- Protection from Traceability attack: It is hard for an attacker to determine whether the same device is sending two distinct authentication requests.
- Protection from malicious gNB_S or gNB_R: It assures that the attacker is unable to retrieve the previous credentials even if the physical access of either gNB_S or gNB_R is seized.

Tables 5.1 and 5.2 specify the notions and acronyms used in the presented description regarding our proposed protocol throughout this chapter.

Table 5.1: Main Notions and Acronyms with their Definition/ Description I

Acronym	Definition	Description
MEC Specific		
gNB	gNodeB	MEC enabled 5G New Radio Base Station
UE	User Equipment	The apparatus that is interfacing to the mobile network on behalf of the user
g2g	gNB-to-gNB	A connection between two gNBs
SMC	Service Migration Channel	g2g channel that is employed for service migration process
MES	Mobile Edge Service	The MEC service instance running in the MEC platform to cater services to the UE
gNB_S	Source gNB	gNB where the MES is currently running and commencing the service migration process
gNB_R	Roaming gNB	gNB that the MES is intended to migrate
TTP/AAA	Trusted Third Party/ AAA Service	This is an Authentication, Authorization, and Accountability (AAA) service formed to conduct identity verification for intended migrations. This entity acts as the governing entity for migration authentications outsourced by the OSS
MVR	MES Verification Registry	MES monitoring functionality of the MEC system that tracks the accountability of MESs
MES Specific		
REQ_{MES}	MES Requirements	Minimum required storage (HDD_{min}), processor (CPU_{min}), memory (RAM_{min}), and bandwidth (BW_{min}) resources, or minimum specifications to execute an APP or an MES. REQ_{MES} can be extracted from the running configuration of the MES.
RE_R	Resource Eligibility	Eligibility of the gNB_R in terms of resources (computing/storage/networking) and SMC capacity to launch the MES. RE_R indicates whether the considered MES has satisfied the REQ_{MES} to launch the service or not.
ID_{MP}	Migration Process ID	Represent a migration related to a single MES at a certain instance.

Table 5.2: Main Notions and Acronyms with their Definition/ Description II

Acronym	Definition	Description
MES Specific		
$DATA_{MES}$	MES Information/ Specifications	The parametric information of a certain MES such as ID_{MES}, $Name_{MES}$, $ID_{Container-MES}$, $OS_{Container-MES}$, $IP_{MES-SERVER}$, and QCI_{MES}
$STATE_{MES}$	MES Status	The running status of the MES that indicates the currently consuming processor and memory configuration to be conveyed to the gNB$_R$, for allocation of resources. gNB$_R$ will allocate the memory and processor resources in accordance with these factors.
Security Specific		
SP	Security Profile	A template that specifies the different security features and parameters applied for a channel
ECC	Elliptic Curve Crypto [158]	A cryptographic method modeled based on the arithmetic of elliptic curves
Enc[]	Enc[Payload, K$_X$]	Encryption : payload encrypted with the key K, either belonging to party X, or symmetric
Sig[]	Sig[Payload, K$_Y$]	Signing: Signed with the private key of party Y
PuK_X	Public Key of X	Used for encryption and unsigning or verifying
PrK_X	Private Key of X	Used for decryption and signing
SyK_{XY}	Symmetric Key of X and Y	Employed for encryption and decryption
n$_X$	Nonce	Nonce generated by party X
P_X	ECC Point	A point on an elliptic curve computed from the random secret $a \in Z_n$, that takes the form $P_X = a.M$, where M is a public elliptic point
$H[input]$	Hash	Hash value of the '$input$'. Typically SHA-256 hashing algorithm is used for this process
MIH	Message Identification Header	Message Identification Header is the application layer message naming/identifying tag, that is standardized in accordance with the protocol
TS	Timestamp	The current timestamp of the system triggered by an event

5.3 Security Protocol Design

Due to the distributed nature of the MEC edge computing deployments, and its system level existing at a distance (i.e. in the core network), a federated identity verification mechanism was followed in designing this protocol. In other terms, the identity of an individual entity was verified through multiple parties via multiple means. Moreover, the MEC systems' reliance on the 5G core network components (i.e. especially Access and Mobility Management Function-AMF, Session Management Function-SMF, and User Plane Function-UPF) makes the designing of the protocol flow complicated [16]. As one of the goals of this protocol is to unburden the main MEC entities of the security and authentication concerns, a Trusted Third Party (TTP) can be employed as a provisioning service. The proposed protocol attribute functional and architectural goals in addition to the security goals defined under the sub-section 5.2.3. Therefore, a complete set of goals targeted by this protocol is stated in sub-section 5.3.2.

A holistic perspective of the proposed security protocol and its connections to the MEC architectural components, operational sequence, and functional parameters are illustrated in Fig. 5.3. The entity Mobile Edge Platform Manager (MEPM) is the orchestrator of the edge level, and the Virtualization Infrastructure Manager (VIM) performs the hypervisor function on virtualized resource management. MEHs are the main operational elements of the MEC system, where their ME APPs or MESs launched in the Virtualization Infrastructure (VI) are governed by the Mobile Edge Platform (MEP) entity. The Local Area Data Network (LADN) within the MEH steers the traffic with the assistance of the UPF. Further information on these entities can be found in [15]. The noteworthy entities in the same figure are described or defined below.

Operations Support System (OSS): This is one of the main entities at the MEC system level. According to the ETSI documentation, OSS is responsible for handling the user access authorization and subscriptions with proper distinguishing of the various service types forwarded from UE App Life-cycle Management Proxy (UALCMP) and Customer Facing Service (CFS) portal [38]. As the main authority for authorization in the MEC domain, it is our main assump-

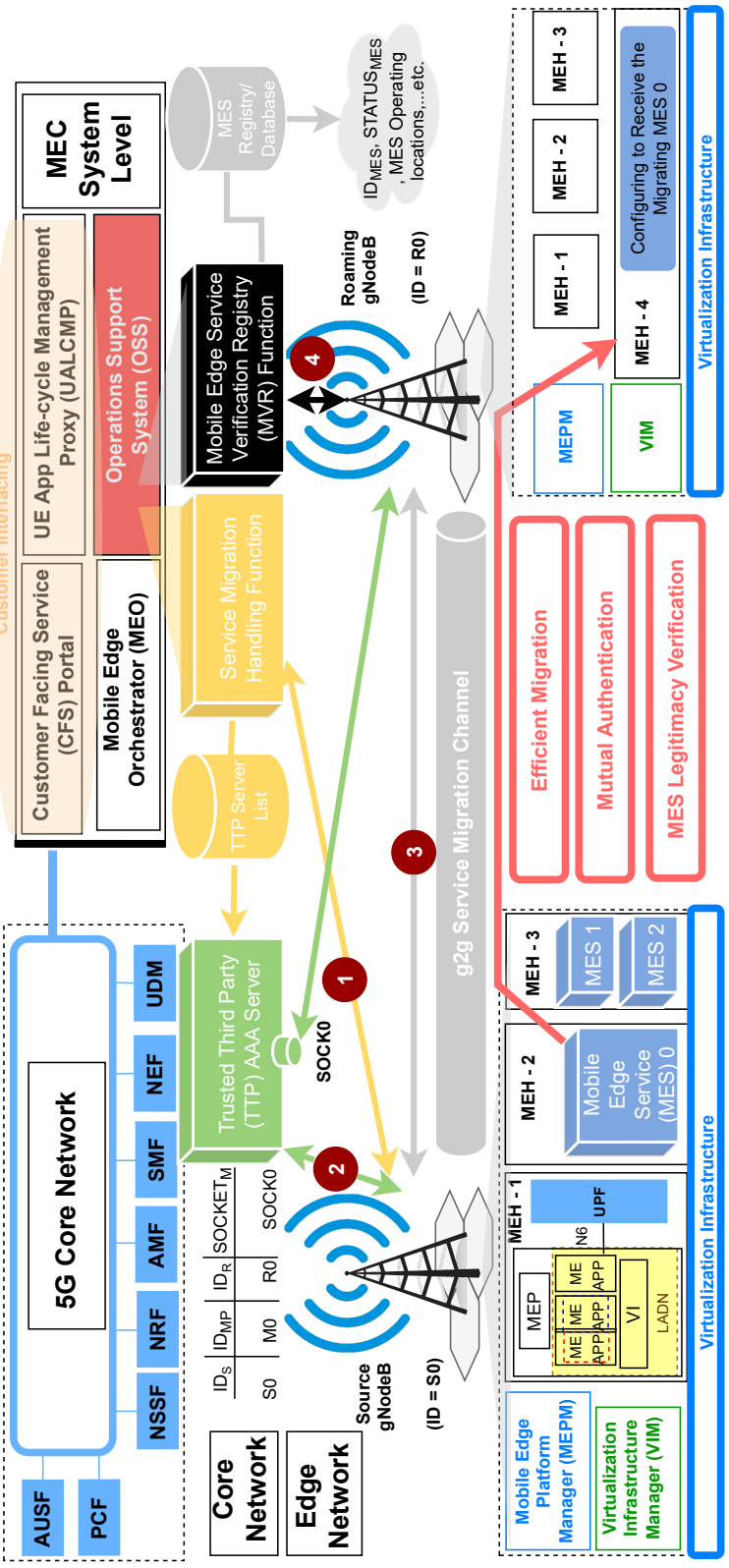


Figure 5.3: Holistic Service Migration Authentication Process from a MEC Architectural Viewpoint

tion that OSS is capable of handling the migration authorizations. Specifically, we are assuming that OSS is catering the functions of 1) serving as the main registry for storing the available Trusted Third Party (TTP) servers tasked with performing Authentication, Authorization, and Accounting (AAA) functions; 2) registry/ database for storing live MES information within the MEC system. There could be many TTP servers as MEC is a distributed architecture. But the gNodeB is considered as the edge level of the MEC. There are several gNBs controlled under a system level.

Purpose of the TTP Server: Due to the allowance in 5G and B5G technologies to launch micro or macro-cell level gNBs from local 5G operators, registering and monitoring all such gNBs under the MEC system registries (i.e. OSS) is not viable, as certain services might be placed locally by their service providers and are launched only for application-specific instances within a limited domain. Due to this reason, there could be fake base stations or fake gNB attacks perpetrated by resourceful adversaries capable of intercepting the 5G radio bands. TTP contrives the required trust domain for the assigned geographical area, specific for migration processes. In fact, TTP offers a Migration Authentication as a Service (MAaaS) which alleviates the burden on the MEC system in regards to handling secure service migrations. If a migration is required by a specific gNB, it should first register under the TTP for the migration. But this registration is only applicable to a single MES migration. However, a single registration can be extended to I g2g migration authentication sessions specific to the considered gNB$_S$ and all the other predicted g_{Ri} locations as indicated in Fig. 5.1.

Mobile Edge Service Verification Registry (MVR): This is an entity we are proposing to act as the global registry for all the MESs operating under the MEC service provider. The MVR is monitoring the service instances of each registered MES across the MEC domain, and updates its registries regarding the status, launched gNB location, and migration status. In the MEC service provisioning environment, physical infrastructure is decoupled from the virtualization domain, and software operations are preferred and dominating. Such a priority given to the softwarized entities can be exploited by perpetrators to induce autonomous constructs within and obscured instilled to the code, and

presents the opportunity to propagate through the virtualized MEC environment with ease. Therefore, in this era, security engineers should treat physical and softwarized entities separately. Hence, even if a gNB is a legitimate entity, a malicious MES could penetrate its MEC environment. Such an MES could prompt a migration request just to propagate its malicious content to other gNBs. Therefore, it is important to verify the legitimacy of each MES that is prompting a migration. Among other tasks related to MESs, MVR is primarily tasked with handling the validation process of the MESs that are requesting a migration.

As specified earlier, MVR can be launched as a functional construct of the OSS in the MEC environment. When migration is prompted by an MES in the gNB$_S$ towards gNB$_R$, an MES verification request (MES_VER_REQ) is sent from the gNB$_R$ to the MVR that embeds the corresponding identities of MES (ID_{MES}), gNB$_R$ (ID_R), and gNB$_S$ (ID_S). The MVR would check its entries for the provided ID_{MES}, and whether that particular MES is registered under the ID_S. If yes; a verification code $CODE_{MES}$ is sent to the gNB$_R$ while the same $CODE_{MES}$ is forwarded to the gNB$_S$. Thus, both gNB$_R$ and gNB$_S$ can attain the verification code and validate the MES legitimacy.

5.3.1 Assumptions

- A1 - Assuming that authentication takes place prior to initiating migration in the pre-migration stage.
- A2 - Assuming that possible gNB_Rs are already selected and known in terms of their network addresses.
- A3 - Assuming that the most suited (e.g. through prediction based on distance) gNB is selected as the 1st gNB_R in the sequence of 1-to-N authentication sessions.
- A4 - Assuming that all the MESs should be pre-registered under the OSS, MVR, UALCMP, or CFSP.

- A5 - All the 1-to-N authentication sessions are occurring independently and in parallel to each other, where gNB_S is equipped with a sufficient number of 5G NR interfaces.
- A6 - Assume that all the links directed from gNB_S have sufficient and a dedicated BW to convey the messages relating to the authentication protocol so that maximum security measures can be applied.
- A7 - Assume that re-transmission protocols of the L2 and L4 of the TCP/IP stack are performing independently to detect packet losses or errors during transmission.
- A8 - Assume the entire MEC system is synchronized and Timestamp (TS) errors are negligible. Realistically, synchronization is a critical issue, especially at μs precision. An undetected synchronization failure will lead to an incorrect clock skew setting that can be exploited by the adversaries targeting Replay attempts. Since the proposed protocol employs TSs for validation, the entire protocol will be disrupted at the initial stages. For the proposed security protocol, however, to protect against Replay attempts, a clock skew at ms level is acceptable. In a MEC system, time offset can be minimized between adjoining gNBs and the connections to the core network due to proximate placement and faster core network link mediums. In addition, 5G presents various solutions to synchronization through its RAN features of faster scheduling, short/robust transmissions, faster retransmissions, preemption, and packet duplication [162]. Thus, real-time continuous time alignment and clock-skew reduction are assurances given for this assumption [163].
- A9 - The OSS contains the list of all the trusted TTP registered under the MNO.
- A10 - $SOCK_{TTP}$ and $SOCK_{TTP_M}$ are having different port numbers, hence the corresponding services can operate independently and simultaneously.

- A11 - Assume that gNB_S and gNB_R are aware of all the public certificates of engaging entities except the TTP.
- A12 - For every protocol session/ segment, all the nonces, timestamps, and HMACs are newly generated and the notions are only bound to the specified protocol session.

5.3.2 Goals of the Proposed Security Protocol

The main goals of the proposed protocol are mentioned below.
G1 – Mutually authenticate each gNB$_R$ selected to initiate the migration with gNB$_S$ in pre-migration stage.

G1.1 – Authenticate gNB$_R$ to gNB$_S$

G1.2 - Authenticate gNB$_S$ to gNB$_R$

G1.3 – Form an identity verification mechanism for gNB$_S$

G2 – Mitigate possible DoS or DDoS attempts on server interfaces of the proposed system model

G3 – Validate the legitimacy of each migrating MES

G3.1 – Propose a governing and monitoring entity for MESs under the MEC system

G3.2 – Propose a method to authenticate and validate the MESs to the governing entity

G4 – Evaluate the Eligibility of gNB$_R$ to host the MES

G5 – Propose a secure SP selection process

G5.1 – Create a secure migration master key K_M, for migration session establishment

G5.2 – Propose a secure method to share available SPs and to conduct the selection process

G6 – Migration session establishment

G6.1 - Integrate the SP to the migration session

G6.2 - Propose a method to migrate the executing MES

5.3.3 Proposed Security Protocol

In the protocol segments described below, several methods are followed to ensure the mutual-authentication among the entities involved in communication. Such methods can be specified as:

- Public Key Encryption: The common RSA encryption based on X509 certificates is followed in this protocol. As these entities are either centralized servers or edge server entities that provide services, we assume a certificate as a viable possession. In convention, asymmetric cryptography is performed with a key pair (i.e. public and private). PuK is employed for encryption while PrK is employed for decryption. All the messages contain encrypted content. Each encrypted content embeds a TS to prevent Replay or Relay threats.

- Elliptic Curve Cryptography (ECC): To preserve the perfect forward secrecy, we employ the ECC in the proposed protocol. An elliptic curve $E_m(c, d)$ is defined as $y^2 = x^3 + cx + d(4c^3 + 27d^2)$ over the finite field F_m; wherein m and n are two large primes, and G is the subgroup of the additive group of points $E_m(c, d)$ with order n and $M \in G$[164]. The following are the three mathematical computational problems of ECC.

  1. Elliptic-curve discrete logarithm (ECDL) problem: It is impossible for the attacker to obtain c, from given cM, where $c \in Z_n^*$ and $Z_m^* = \{1, 2, ...m - 1\}$.
  2. Elliptic curve-computational Diffie-Hellman (ECCDH) problem: Even if attacker knows cM and dM then it is impossible for the attacker to compute cdM, where $(c, d \in Z_m^*)$.
  3. Elliptic curve-descisional Diffie-Hellman (ECDDH) problem: Attacker can not guess the $em : em = cdm$, even if he has cM, dM, and eM where $(c, d, e \in Z_m^*)$.

- Timestamp Utilization and Freshness Verification: TSs are widely used in security protocols for assuring freshness. TS mechanisms are vital for detecting Replay or Relay attempts, and an unintentional performance

indicator. Freshness is assured by verifying whether the received TS from the message is within the defined clock skew of the system. In fact, if TS_r, TS_c, ΔT represents received, current timestamps, and the allowable clock skew: $(TS_c - TS_r) < \Delta T$. In the proposed protocol, every message is TS based, and its freshness is determined by detecting whether the TS is within the clock skew at the receiving end. For successful integration of TSs within a system, every networked entity in the system should be synchronized. We are assuming A8 in subsection 5.3.1, in designing our protocol.

- Hashing Functions: Hashing is a technique adopted to create a unique outcome for a specific input, where collisions among the outcomes are very rare. In order to achieve this trait, a one-way trap-door function is utilized as the hashing algorithm. For the proposed protocol, we assume the hashing algorithm is SHA-512.

- Nonces and Nonce Verifying Hashes: Nonces are popular randomized fresh values employed in authentication mechanisms that induce dynamism and liveliness to a protocol. In fact, nonce-based verification is a Challenge-Response (CR) based mechanism. In this protocol, various nonces are employed by all the parties, and a specific hashed nonce (i.e. $H[n_X||TS]$) is sent back by the received party to validate the nonce challenge.

- Signature: Employed to validate the authenticity of the communicating party. The signatures are created with the PrK of the signing party, where anyone in possession of the PuK of the signed party can verify it. The signed content typically contains the entity ID and the relevant TS. Though, signed content is hashed to prevent its exposure to the listeners. Once the hash value is validated (i.e. if the received hash equals the re-computed hash) and the TS is within the defined clock skew, the signature can be considered legitimate.

- Hashed Message Authentication Code (HMAC): HMACs are integrity protection mechanisms commonly employed in protocols. Typically a

hash of the message content is computed and appended to the same message to detect any tampering has occurred. In this protocol, HMACs intend to detect any tampering with the message content, by doing so ensure the integrity of the messages.

- DoS Puzzle: For server interfaces, DoS or DDoS threats are highly probable. Thus, a mechanism should exist to enforce the request sender to perform a certain level of computational task, as a CR mechanism. The DoS puzzle employed in this chapter follows the initial design proposed by [165], and adopted in [158] for edge-based server engagements. The puzzle specifies the k_{DoS} parameter, which represents the complexity. If the Server/ Client IDs and nonces are denoted by ID_S, ID_C, n_S, and n_C, and $H[]$ is a hash function; The puzzle is $H[ID_S||ID_C||n_S||n_C||X] = 0_1 0_2 0_3 ... 0_{k_{DoS}} Y$
The challenge is to determine the X value and convey it to the server for validation. The implementation details and the performance of the DoS Puzzle are further explained in Appendix A.4.

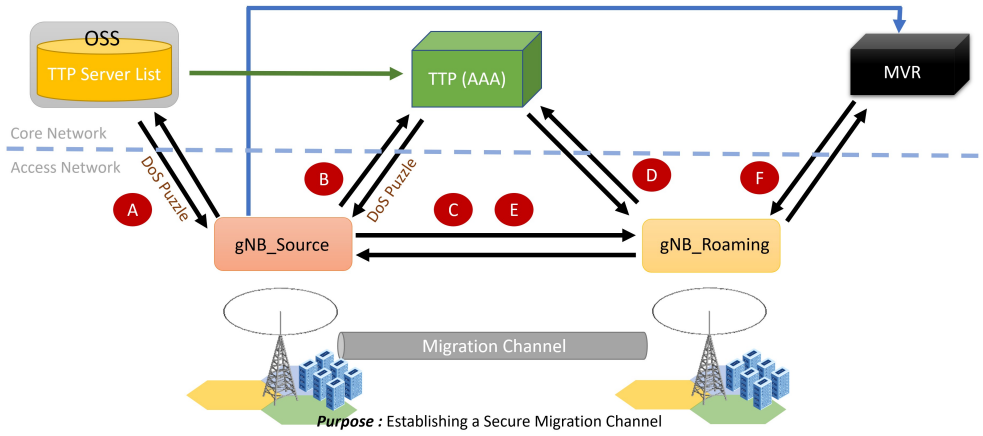


Figure 5.4: High-Level Illustration of the Proposed Protocol

Fig. 5.4 indicates the high-level view of the g2g authentication protocol for migration. The process initiates with gNB_S reaching out to the OSS in step (A) for requesting the contact details of the TTP entity assigned to the MEC domain the gNB_S is operating. With the received TTP credentials, gNB_S will

then contact the migrating AAA service as shown in step (B). TTP will register the respective migrating request and create a unique link/socket specific to this migration, while session IDs are created for the same process. With these credentials, gNB_S will reach out to gNB_R in stage (C). gNB_R will then utilize the unique socket/link to access the TTP server (i.e. in (D)) and verifies the request forwarded from the gNB_S; hence mutual authentication is established. The information regarding the MES and resource requirements are forwarded to the gNB_R by gNB_S in step (E). gNB_R verifies the legitimacy of the MES via MVR (i.e. in (F)) and investigates the resource capability to host the MES, while a migration master key is contrived and shared through the SMC. Thus, this concludes the protocol with the selected SP is in compliance with both parties. The following sub-sections are discussing each section of the protocol extensively. The current development of the protocol only focuses on authentication prior to migration session creation and assumes A5 and A6.

5.3.3.1 Part A: gNB$_S$ to OSS Communication for Acquiring the TTP Credentials

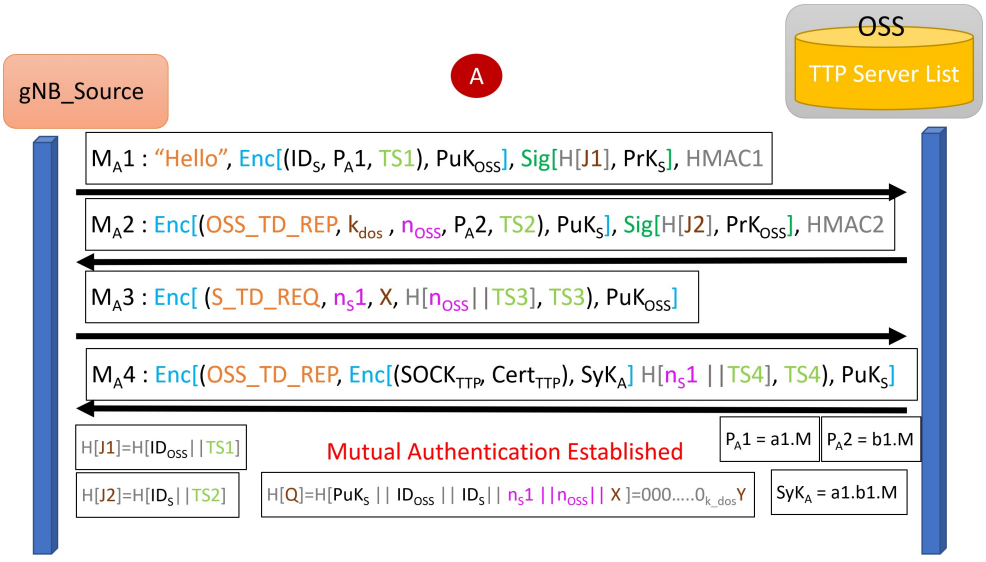


Figure 5.5: Part A of the Proposed Security Protocol that takes place between gNB$_S$ and the OSS

In order to communicate with the TTP, the gNB$_S$ should first acquire the contact information of the relevant TTP assigned for the geo-domain. Assuming A11, gNB$_S$ initiates the communication with OSS. The protocol flow of this segment A is illustrated in Fig. 5.5. In the first message M_A1, gNB_{Source} chooses a random number $a1$ ($a1 \in Z_n$) to compute $P_A1 = a1.M$, where M is an ECC public point. Along with P_A1, "Hello" MIH, ID_S (i.e. Identity of the gNB$_S$ in the MNO network), and $TS1$ are embedded and encrypted with the PuK_{OSS}. In the same message, the content $J1 = ID_{OSS}||TS1$ is hashed signed by PrK_S, while the $HMAC1 = H[ID_S||P_A1||ID_{OSS}||TS1]$ is appended. At the OSS end, ID_S is browsed in the gNB registry, while the freshness of $TS1$ and $HMAC1$ validity is inspected. The OSS then selects relevant MIH, k_{DoS}, compute $P_A2 = b1.M$ with the randomized selection of $b1$ ($b1 \in Z_n$), and generate n_{OSS} to be included inside the encrypted envelop. The $H[J2] = H[ID_S||TS2]$ is signed by the PrK_{OSS}, and $HMAC2 = H[k_{DoS}||n_{OSS}||ID_S||P_A2||TS2]$ is computed to form the M_A2 along with the encrypted content. Upon receiving M_A2, gNB$_S$ extract k_{DoS}, $P2$ and n_{OSS} through decryption, while verifying OSS signature, $TS2$, and $HMAC2$.

After generating n_S1, gNB$_S$ computes the puzzle using $H[PuK_S||ID_S||ID_{OSS}||n_S1||n_{OSS}||X] = 0_1 0_2 ... 0_{k_{DoS}} Y$. With X determined, the gNB$_S$ composes M_A3 with the MIH, $n_S1, X, H[n_{OSS}||TS3]$, and $TS3$ within the encryption. Since signatures are already verified, they are no longer required. Upon receiving M_A3, OSS conducts $TS3$ and OSS nonce verification while recomputing the puzzle utilizing X to verify the compliance on the complexity parameter. After validating the identity of the gNB$_S$ and detecting the request as a non-DoS attempt, OSS forwards the TTP SOCKET (i.e. combination of IP address and the port number of the AAA server), TTP Certificate, along with the n_S1 nonce verification to the gNB$_S$ encrypted with PuK_S in M_A4. As $SOCK_{TTP}$ and $Cert_{TTP}$ are the selected secrets on this protocol segment, we employ light-weight AES encryption as a double-encryption ploy, where the key deriving parameters were exchanged through the ECCDH method to ensure PFS. Thus, two secrets are encrypted with AES using the key $SyK_A = a1.b1.M$. Once received at the requester

end, decrypted, and validated, mutual authentication is established between gNB$_S$ and OSS. In fact, mutual authentication is validated by means of signatures, nonces, and timestamps. Though this is a singular connection, mutual authentication is vital to extend the trust domain to the MEC system level. Though it is not indicated in Fig. 5.5, the OSS function notifies the TTP server of the gNB$_S$ request anchored through ID_S as in Fig. 5.4. We assume that this communication is secure as this is extended within the MEC system-level trust domain.

5.3.3.2 Part B: TTP and the gNB$_S$ communication for obtaining the TTP link for Migration Registration at the TTP

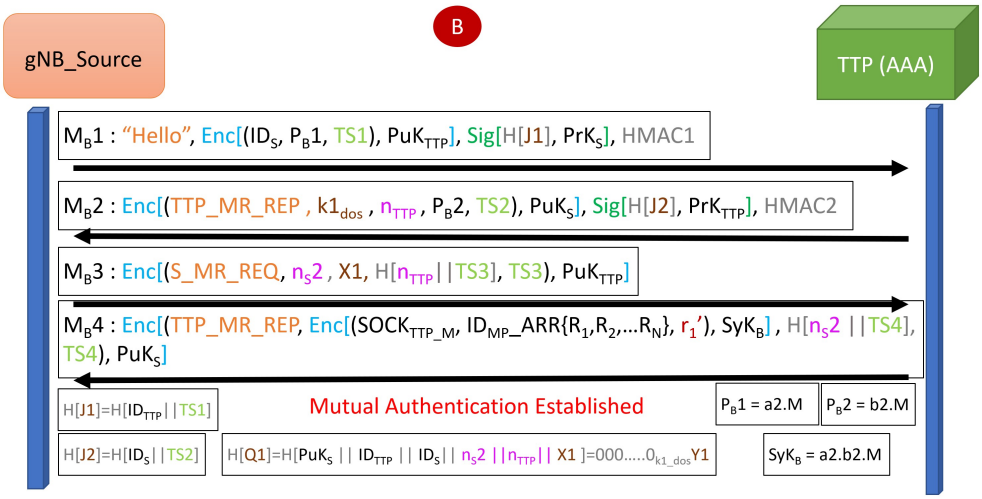


Figure 5.6: Part B of the Proposed Security Protocol that takes place between gNB$_S$ and the TTP

The gNB$_S$ should first register under the migration TTP server so that it will issue the relevant credentials to initiate the migration authentication. This registration phase B is depicted in Fig. 5.6. The request M_B1 is sent by gNB$_S$ to the TTP using the $SOCK_{TTP}$ provided by the OSS, and utilizing the TTP certificate. The initial request M_B1 includes the $ID_S, TS1$, and P_B1 (i.e. $P_B1 = a2.M$, $a2 \in Z_n$) within the encrypted envelop, while the signature formed with the hash $H[J1] = H[ID_{TTP}||TS1]$ and the $HMAC1 =$

$H[ID_S||ID_{TTP}||P_B1||TS1]$ have been embedded to it. Upon receiving this message, TTP will ensure the freshness of the message from the decrypted $TS1$ and verifies the signature to guarantee it was sent by gNB$_S$. The integrity is validated from the $HMAC1$. After all the verification steps, ID_S is put under a temporary migration registration. As TTP represents a server function, for DoS mitigation, $k1_{DoS}$ is selected, P_B2 is computed from the selected $b2$ (i.e. $P_B2 = b2.M$, $b2 \in Z_n$), and n_{TTP} nonce is generated. These three parameters are sent within the encrypted reply of M_B2, along with the corresponding TTP signature that include $H[J2] = H[ID_S||TS2]$ and $HMAC2 = H[k1_{DoS}||n_{TTP}||ID_S||P_B2||TS2]$. gNB$_S$ decrypts the message and performs the relevant checks on the MIH, TTP signature, $TS2$, and $HMAC2$. After they are verified and n_S2 is generated, it will determine $X1$ from the puzzle $H[Q1]$ utilizing $k1_{DoS}$: $H[PuK_S||ID_S||ID_{TTP}||n_S2||n_{TTP}||X1] = 0_1 0_2 0_3 ... 0_{k1_{DoS}} Y1$.

The next message M_B3 from gNB$_S$ includes n_S2 and $X1$ parameters along with the TTP nonce verification $H[n_{TTP}||TS3]$ and timestamp within the encrypted envelop. Once the TTP receives and decrypts M_B3, nonce verification is checked, DoS Puzzle is checked using $X1$, while freshness check follows. The TTP then creates the migration socket $SOCK_{TTP_M}$ and exposes it, where it is specific and unique for the migrations initiated by gNB$_S$. Only gNB$_S$ can access that socket which can be authenticated by ID_S. Then Migration Process Identities (i.e. ID_{MP}s) specific for the gNB$_S$ are generated. These ID_{MP} will be stored in an array: $ID_{MP}ARR(R_i) = [ID_{MP(R0)}, ID_{MP(R1)}, ID_{MP(R2)}, ...ID_{MP(RI)}]$, where each $ID_{MP(Ri)}$ represents the ID_{MP} specific for the relevant migration point g_{Ri} or gNB$_R$ station, while the size of I is dependent on the accuracy and the range of the predicted path of the UE in reference to Fig. 5.1. A random value r_1 is generated and its modular value $r_1' = r_1 mod N$ is sent to the gNB$_S$ along with the $SOCK_{TTP_M}$, and $ID_{MP}ARR(R_i)$ encrypted by AES employing SyK_B (i.e. $SyK_B = a2.b2.M$); along with the hashed S nonce $H[n_S2||TS4]$ forming M_B4. Upon receiving and decrypting M_B4, gNB$_S$ will store the received information for further processing in the next stages. At this point, mutual authentication is established between the gNB$_S$ and the TTP.

5.3.3.3 Part C and D: gNB$_S$ to gNB$_R$ initial communication prior to MES verification

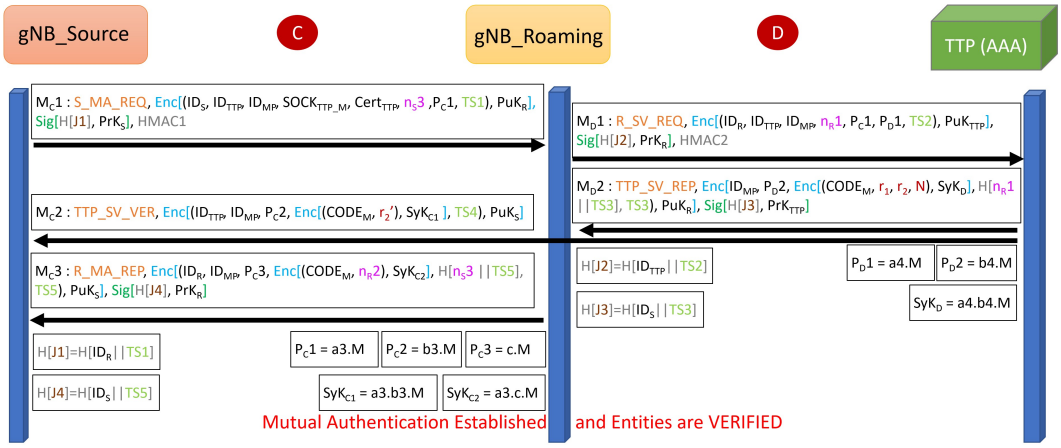


Figure 5.7: Part C and D of the Proposed Security Protocol that takes place between gNB$_S$, gNB$_R$ and the TTP

In this stage as illustrated in Fig. 5.7, both gNB$_S$ and gNB$_R$ entities are verified to each other leveraging the TTP connectivity shared through the $SOCK_{TTP_M}$. gNB$_S$ is initiating the migration request towards gNB$_R$ from M_C1, including $ID_S, ID_{TTP}, ID_{MP(R0)}, SOCK_{TTP_M}, Cert_{TTP}, n_S3$, $TS1$, and P_C1 (i.e. $P_C1 = a3.M$, $a3 \in Z_n$) within the encrypted envelop. In addition, gNB$_S$ signature embedding $H[J1] = H[ID_R||TS1]$, and $HMAC1 = H[ID_S||ID_{TTP}||ID_{MP(R0)}||SOCK_{TTP_M}||Cert_{TTP}||n_S3 ||ID_R||P_C1||TS1]$ generated with all the stats in the encrypted and signature envelops are appended. The received M_C1 at the gNB$_R$ is decrypted at first while signature, freshness, and $HMAC$ validations are carried out. Then gNB$_R$ utilizes the $SOCK_{TTP_M}$ to establish the specific connection to TTP, and formulates the encrypted envelop using $Cert_{TTP}$ including $ID_R, ID_{TTP}, ID_{MP(R0)}, n_R1, P_C1, TS2$, and P_D1 (i.e. $P_D1 = a4.M, a4 \in Z_n$). Since the connection information was shared by an outside party, verifying the identity of the TTP is vital for gNB$_R$. Thus, both a nonce and the gNB$_R$ signature $H[J2] = H[ID_{TTP}||TS2]$ are embedded in M_D1, in addition to $HMAC2 = H[ID_R||ID_{TTP}||ID_{MP(R0)}||n_R1||P_C1||P_D1||TS2]$.

Upon receiving the message M_D1 from gNB$_R$ through the exposed socket

$SOCK_{TTP_M}$, the message will be decrypted and the usual violation detection/ verification schemes are carried out. With the received information, $ID_{MP(R0)}$ is anchored into the TTP migration registry for locating ID_S and cross-checking with ID_{TTP}. Then the ID_R is temporarily registered for a possible migration. The migration code is generated $CODE_M = H[ID_S||ID_R||ID_{MP(R0)}||n_{RAND}]$ to verify the migration registration with all the parties. n_{RAND} is a random nonce known only to TTP. This $CODE_M$ is disseminated to both gNB$_R$ and gNB$_S$ with the messages M_D2 and M_C2 respectively. The M_D2 include the $ID_{MP(R0)}$ to improve the convenience over the browsing through the migration registry, and the generated $CODE_M$. TTP performs several computations to select the values r_2 and r_2', where r_1 and r_1' are already available, where $r_2' = r_2 mod N$. The r values are selected as $[r_1, r_1', r_2, r_2'] : (r_1, r_2 > N) \wedge (r_1' \times r_2' < N)$.

The M_D2 contain r_1, r_2, N, and $CODE_M$ within the AES encryption generated from SyK_D (i.e. $SyK_D = a4.b4.M$); in addition to the $ID_{MP(R0)}$, P_D2 (i.e. $P_D2 = b4.M$, $b4 \in Z_n$), hashed nonce $H[n_R1||TS3]$, and the TTP signature. With this reply successfully received and verified, gNB$_R$ is ensured of the legitimacy and trustworthiness of TTP. In M_C2, $CODE_M$, and r_2' are AES encrypted with SyK_{C1} (i.e. $SyK_{C1} = a3.b3.M$), while $ID_{TTP}, ID_{MP(R0)}, P_C2$ (i.e. $P_C2 = b3.M$, $b3 \in Z_n$), and $TS4$ are conveyed additionally. After M_D2, gNB$_R$ compiles a reply to the gNB$_S$ including $CODE_M, n_R2$ within the AES encrypted envelop created by SyK_{C2} (i.e. $SyK_{C2} = a3.c.M$); and $ID_R, ID_{MP(R0)}$, P_C3 (i.e. $P_C3 = c.M$, $c \in Z_n$), $H[n_S3||TS5]$, and $TS5$ contained in the RSA encrypted envelop, while gNB$_R$ signature is also appended in M_C3. The received two $CODE_M$s will be cross-checked at the gNB$_S$ end with the received information. This phase of the protocol concludes with registering the migration under the TTP entity and establishing mutual authentication among gNB$_R$ and gNB$_S$. Further, AES-based double-encryptions conducted from SyK_D, SyK_{C1}, and SyK_{C2}, which were computed from the factors disseminated by ECCDH means, ensure the PFS for sensitive credentials of the protocol.

5.3.3.4 Part E and F: MES Verification by MVR to improve the Trust domain

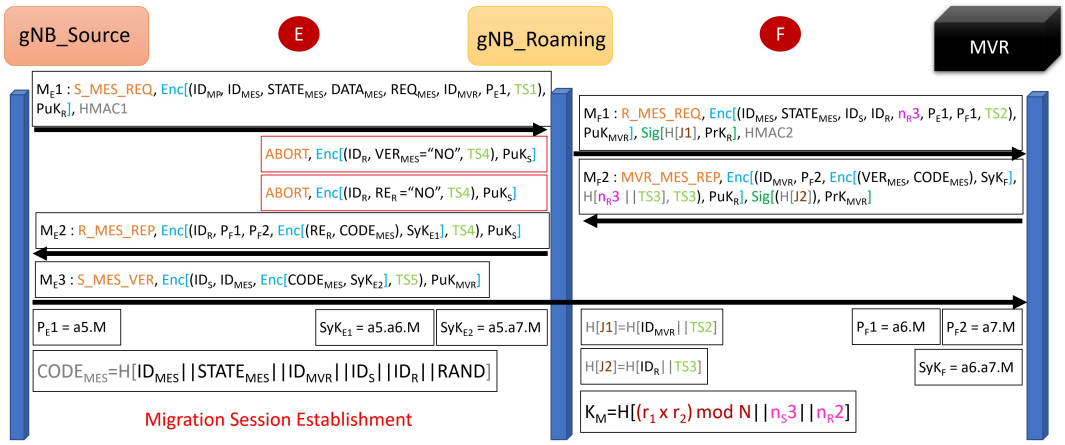


Figure 5.8: Part E and F of the Proposed Security Protocol that takes place between gNB$_S$, gNB$_R$ and the MVR

One of the main requirements of this authentication protocol is the validation of the MESs that are intended to be migrated, as specified in G3. This portion of the authentication protocol depicted in Fig. 5.8 deals with that requirement, where the MVR entity provides the intrinsic validation logic towards the MEC system. In addition, the gNB$_R$ is investigating whether the intended MES has sufficient resources to launch the service in its virtualization environment. Upon receiving the migration verification code $CODE_M$ from the gNB$_R$, gNB$_S$ forms a message including $ID_{MP(R0)}, ID_{MES}, STATE_{MES}, DATA_{MES}, REQ_{MES}$, ID_{MVR}, P_E1 (i.e. $P_E1 = a5.M$, $a5 \in Z_n$), and $TS1$ within its encrypted envelop and appends the integrity measure $HMAC1 = H[ID_{MP(R0)}||ID_{MES}||STATE_{MES}||DATA_{MES}||$ $REQ_{MES}||ID_{MVR}||P_E1||TS1]$. Further details on these indexes are specified in Tables 6.1 and 6.2. The gNB$_R$ is contacting the relevant MVR server function (i.e. operating under the OSS entity) leveraging the ID_{MVR}. In the first contact message M_F1, gNB$_R$ include $ID_{MES}, STATE_{MES}, ID_S, ID_R, n_R3, TS2$, P_E1, and P_{F1} (i.e. $P_F1 = a6.M$, $a6 \in Z_n$) along with the gNBR signature and $HMAC2$. The MVR after decrypting the received message and vali-

dating/ verifying, the ID_{MES} is anchored to seek the ID_S in its database. If the respective ID_{MES} is bound to ID_S, ID_R is temporarily registered in the MVR registry as an MES user. Since the MES claim is legitimate, a code is generated indicating the validation denoted by $CODE_{MES} = H[ID_{MES}||STATE_{MES}||ID_{MVR}||ID_S||ID_R||n1_{RAND}]$. $n1_{RAND}$ is a random nonce similar to the generation of $CODE_M$. The verification status of the MES is indicated by VER_{MES} (i.e. either YES or NO). The MVR then composes the reply M_F2 with VER_{MES} and $CODE_{MES}$ within the AES encryption envelop created from SyK_F (i.e. $SyK_F = a6.a7.M$); and include ID_{MVR}, P_F2 (i.e. $P_F2 = a7.M$, $a7 \in Z_n$), $H[n_R3||TS3]$, and $TS3$ within the RSA encrypted envelop along with the MVR signature.

Upon reception, decryption, and validation of M_F2, gNB$_R$ stores the $CODE_{MES}$. If the verification value is VER_{MES} = NO, the process will be aborted and the TTP will be notified. If VER_{MES} = YES, the resource availability check is conducted in accordance with the procedures specified in [30]. If the available resources in the gNB$_R$ are sufficient to host the MES (i.e. REQ_{MES}), then the entity proceeds to the next step, and the indicator RE_R = YES. gNB$_R$ then convey the message M_E2 embedding $RE_R, CODE_{MES}$ within a AES encrypted envelop created from SyK_{E1} (i.e. $SyK_{E1} = a5.a6.M$); and appending ID_R, P_F1, P_F2, and $TS4$ inside the RSA encryption. If RE_R = NO, the process is aborted and notified to both gNB$_S$ and TTP. After sending the final message, gNB$_R$ computes the shared migration master key, $K_M = H[(r_1 \times r_2 mod N)||n_S3||n_R2]$.

After receiving the RE_R and $CODE_{MES}$, gNB$_S$ computes the same migration master key, $K'_M = H[(r'_1 \times r'_2)||n_S3||n_R2]$. Due to the properties of modular arithmetic, $K_M = K'_M$ and can be used as the master configuration key for the migration session. It can be noted that N is never sent to gNB$_S$, and TTP is unaware of the values n_S3 and n_R2. Hence, both gNB$_S$ and gNB$_R$ are computing the K_M in different means, while TTP or any other entity is unaware of all the required values. After initiating the migration process, gNB$_S$ notifies the MVR composing M_E3, with $CODE_{MES}$ encrypted with AES key SyK_{E2} (i.e. $SyK_{E2} = a5.a7.M$), and including ID_S, ID_{MES}, and $TS5$ to notify and make the gNB$_R$ registration of ID_{MES} permanent.

5.3.3.5 Part G: Migration Session Establishment

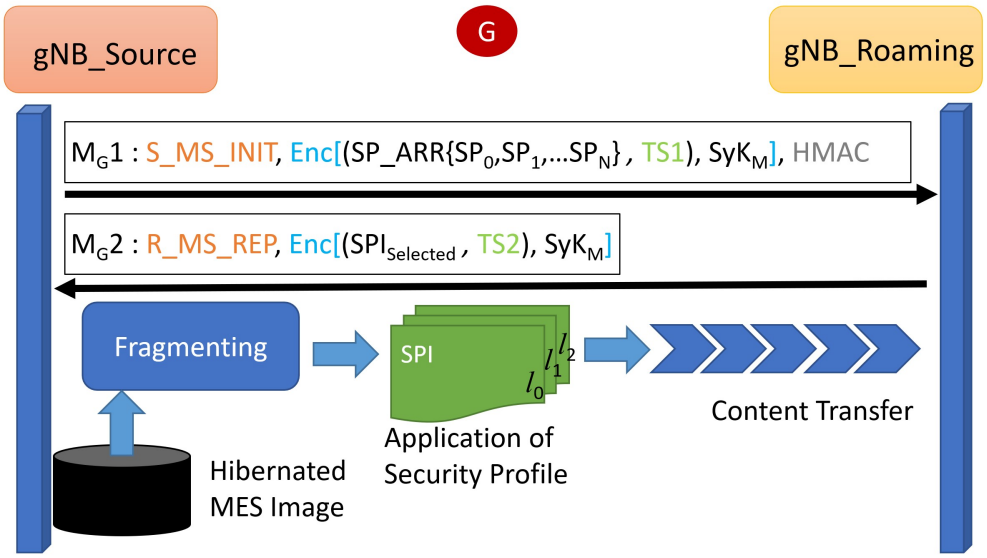


Figure 5.9: Migration Session Establishment Phase of the Proposed Protocol

As specified in Subsection 5.3.2, and the goals G5 and G6, the proposed authentication protocol concludes in a situation where the most suited SP can be applicable to the transferring of the migrating content. Thus, this stage can be considered as the migration session establishment phase of the protocol as illustrated in Fig. 5.9. Since K_M is already determined, an AES-512-based symmetric key is generated utilizing K_M (i.e. K_M is used as the secret key spec), which forms the SyK_M. This SyK_M is intended to be employed in signaling message transfers during the migration session. At the initiation, M_G1 is sent from gNB$_S$ to gNB$_R$ composing all the available SPs that are acceptable to gNB$_S$. The M_G1 is encrypted by SyK_M while a $HMAC$ is appended, as the integrity of this message is vital for the migration session establishment. Upon receiving, gNB$_R$ will select the most suited SP to initiate the migration considering its computational and bandwidth capability. Hence, the SPI of the selected SP is conveyed to the gNB$_S$ encrypted with SyK_M in M_G2. Since the migration g2g tunnel between gNB$_S$ and gNB$_R$ is established, the containerized MES is hibernated into an image that includes its running configuration. Then the fragmented image content is subjected to the relevant cryptographic

operations specified under the selected SP. The encrypted content is then migrated to the gNB$_R$ MEC environment to be decrypted, assembled, and configured to launch the MES in the new environment.

5.4 Informal Analysis

This section provides the informal or descriptive analysis of the proposed protocol that establishes proofs for the 7 specified propositions.

Proposition 1. The proposed protocols provide Mutual authentication.

Proof. This preposition explains how the proposed protocols (parts A and part C& D) deliver mutual authentication.

- **Proof for Part A:** When gNB_{Source} receives message $((OSS-TD-RP,(SOCK_{TTP},Cert_{TTP})_{SyK_A},$ $H[n_s1,TS4],TS4)_{PuK_s})$ from the OSS. gNB_{Source} decrypts this message and compute the $H[n_s1,TS4]^*$ in order to compare $H[n_s1,TS4]^* == H[n_s1,TS4]$ with the received. If it matches then believe that OSS is authentic because n_s1 was sent using the public key of OSS and OSS only knows. On the other hand, when OSS receives $((S_{TD-REQ},n_s1,X,H[n_{OSS},TS3])_{PuK_{OSS}})$ from the gNB_{Source} then it decrypts the message to obtain the credentials. After decrypting the message, OSS computes $H[n_{OSS},TS3]^*$ and compares with the received $H[n_{OSS},TS3]$ (i.e., $H(n_{OSS},TS3) == H(n^*_{OSS},TS3)$). If it matches then OSS believes that gNB_s is authentic because gNB_{Sourec} only knows the n_{OSS}.

- **Proof for Part C&D:** When gNB_{Source} receives the message $((ID_{TTP},ID_{MP},P_C2,(CODE_M,r_2^{'})_{SyK_{C1}},TS4$ $)_{PuK_S})$ from the TTP and $((ID_R,ID_{MP},P_C3,(CODE_M,n_s2)SyK_{C2},H[n_s2,$ $TS5],TS5)_{PuK_S})$ from the gNB_R. gNB_S decrypts this message and compute the $H[n_s3,TS5]^*$ in order to compare $(H[n_s3,TS5]^* == H[n_s2,TS4],CODE_{M_{gNB_R}} == CODE^*_{M_{TTP}})$. If it matches then believe that gNB_R is authentic because n_s3

was sent using the public key of gNB_R and gNB_R can only know. On the other hand, when gNB_R receives the message $((ID_{MP}, P_{D2}, (CODE_M, r_1, r_2, N_1)_{SyK_D},$ $H[n_R1, TS3], TS3)_{PuK_R})$ from the TTP. After decrypting the message, gNB_R computes $H[n_R1, TS3]^*$ with the stored n_R1 in order to compares $H(n_R1, TS3) == H(n_R1^*, TS3), ID_{MP} == ID_{MP})$ than gNB_R believes that gNB_S is authentic because gNB_S only knows the ID_{MP}.

Thus this shows that all three proposed protocols provide Mutual Authentication.

Proposition 2. The proposed protocols provide Anonymity protection.

Proof. In the proposed protocols (i.e., part A and part C& D), the identities of the gNB_S, gNB_R, OSS, and TTP are exchanged in the encrypted form instead of plaintext over the insecure channel. For the part A, identities of entities involve in communication gNB_S and OSS are transmitted in encrypted and hashed form $(ID_S, P_A1, TS1)_{PuK_{OSS}}$ and $HMAC1 = H(ID_S, ID_{OSS}, TS1)$. For the part $C\&D$, identities of entities involve in communication gNB_S, gNB_R and TTP are transmitted in encrypted and hashed form $(ID_S, ID_{TTP}, ID_{MP}, P_{C1}, SOCK_{TTP}, n_s3, TS1)_{PuK_R}$. Therefore, we can clearly see that even if an attacker eavesdrops or captures the exchanged messages, he will be unable to get the identity of gNB_S, gNB_R, OSS, TTP because they are exchanged in encrypted from using the public key encryption instead of plaintext. Thus, our proposed protocols provide Anonymity protection.

Proposition 3. The proposed protocol provides Perfect Forwards Secrecy.

Proof. The proof of this proposition elaborates that the attacker can not acquire the session key even though he has the private key of communicating entities.

- Part A: If an attacker obtains the private key of gNB_S and OSS (i.e., PrK_S, PrK_{OSS}) then he can not determine the $SOCK_{TTP}$ and $Cert_{TTP}$ because they are encrypted with SyK_A. It is impossible for the attacker to compute the $SyK_A = a1.b1.M$ due to the intractability of ECDL and ECCDH problem [166].

- Part C & D: If an attacker obtain the private key of gNB_S, gNB_T and TTP (i.e., PrK_S, PrK_R, PrK_{TTP}) then he can can not determine the $CODE_M, r_1, r_2, N$, $CODE_M, r_2'$ and $CODE_M, n_R2$ because they are encrypted with $SyK_{C1}, SyK_{C2}, SyK_D$. It is impossible for the attacker to compute the $SyK_{C1} = (a3.b3.M), SyK_{C2} = (a3.C), SyK_D = (a4.d4)$ due to intractability of ECDL and ECCDH problem [166].

Note: Similar to PFS providing secrecy for past conversations by securing the cipher key, the concept called 'backward secrecy' is defined to entail the property of secrecy of the future content to be conveyed through the protocol [167]. This is a measure introduced to extend the security even if the PFS is compromised, specifically in group-communication scenarios where the session time of a cipher might be extended due to the higher cost of session key re-establishment. Though the MEC-SMAP protocol proposes a 1-to-N authentication model, each authentication session is operated individually. Thus, it is not a group-communication scenario where backward secrecy is vital. Hence, the design and formulation of MEC-SMAP omitted the implementation of backward secrecy.

Proposition 4. The proposed protocol is resilient against the Replay attack.

Proof. To assure the replay attack protection, we employ the timestamp and nonce in each message exchange through which communicating parties gNB_S, gNB_R, OSS, and TTP could verify the freshness of the exchanged message. For, e.g., in Part A, when OSS receives the $((ID_S, P_A1, TS1)_{PuK_{OSS}}, Sig[H(J1)]_{PrK_S}, HMAC1)$ then first it verifies the freshness of the exchanged message by checking the freshness condition $(TS_c - TS_r < \Delta T)$ (i.e., $TS2 - TS1 \leq \Delta$). If it holds, then it accepts the message and decrypts the message. Otherwise, abort the process. The same approach is applied for the rest of the message exchanges for part A, and part C& D to verify their freshness by the receiving end. Hence, the proposed protocols are resilient against Replay attacks.

Proposition 5. The proposed protocols are resilient against the Denial-of-service (DoS) attack.

Proof. In the proposed protocols, we use the timestamp and nonce to exam-

ine the freshness of the message (i.e., the message was not sent previously). All the communication entities such as gNB_s, gNB_R, OSS, and TTP of Part A, B, C& D verify the freshness of the message by checking the freshness conditions. If the freshness condition meets, then they again verify the timestamps after verifying the signature of the message. If the timestamp is found correct in both checks, then only the message is accepted. Otherwise, they reject the message and abort. The analysis shows that the attacker can not replay the captured message due to the proper use of timestamps and nonce. Therefore, it is hard for an attacker to launch a denial of service attack. Hence, the proposed protocol is resilient against the DoS attack.

Proposition 6. The proposed protocols are resilient against traceability attacks.

Proof. This attack is impossible due to the usage of random numbers, timestamps, and nonces which are changed after each successful authentication request. The random numbers, timestamps, and nonce utilized in two separate sessions are completely unrelated to one another. Assume the attacker obtains a copy of the messages exchanged between the several sessions. In such a situation, he will not be able to connect communications from one session to messages from another since each session's signature and authentication replies are generated using fresh random numbers, nonce, and timestamps. Consequently, an attacker is unable to link the messages of one session to those of another. As a result, the proposed protocols are resistant to traceability attacks.

Proposition 7. The proposed protocols provide protection from malicious gNB_S and gNB_R.

Proof. Malicious gNB_S or gNB_R is meant by the physical compromise of the entity during its operation. The nature of service delivery in the MEC system, as explicated in [25], is dependent on the virtualization platform that extends from the edge to the core of the network towards its system level. In fact, operations of the MEC system decouple the physical and virtual domains. The governing entity of the MEC edge level, the Mobile Edge Platform Manager is constantly communicating with the Mobile Edge Orchestrator and the rest of the system-level entities to decide on the inception, operation, and termination

of services; while Virtualization Infrastructure Manager controls the resource allocations and manipulations depending on the service requests. Thus, this autonomous environment is operating without any network administrator at the edge level, while all the governing decisions are conveyed from the system level to the edge through the virtual channels. Hence, access to the MEC edge-level servers or entities cannot be gained by an intruder who has physical access to the system. In fact, an interface is not required for administrative access at the edge, other than a monitoring terminal. Thus, MEC-enabled gNBs are secured against physical threats to the system by design. The reliance on the system level for maintaining the operations, however, induces a possibility for attackers to block the communication channels between the system and edge level, which would isolate the gNB. Therefore, the security of this edge-core channel is a prime requirement for MEC deployments.

Informal analysis has been done for part A and part C& D since part A is identical to part B and part C&D is identical to part E&F. Therefore, for B and for part c & D, we can follow the same informal analysis approach as part A and part C& D respectively.

5.5 Formal Analysis

This section presents the formal analysis of the proposed protocol using the GNY logic, ROR logic, Scythe tool [168], and the AVISPA tool.

5.5.1 Model Based Verification with Scyther

The Scyther tool was employed to verify the proposed protocol for its resiliency, as it is a well-known automated tool for validating security protocols following a model-based approach. The Security Protocol Description Language (SPDL) is used to specify the protocols in Scyther, where testing protocol segments are specified under spdl files. Since the proposed protocol is described under the segments A, B, C, D, E, and F, the specifications were conducted under 4 SPDL files. A and B segments were specified under two different files, while C & D, and E & F segments were specified with 3 roles.

In the tool, verification, advanced, and graph output parameters are controlling how the validation output is conveyed. Under verification, the maximum number of runs was set to 100 for every parameter sequence in the protocol that was tested. In the first sequence, the matching type was set to *find basic type flaws* while search pruning in advanced parameters was set to *find all attacks*. In the second sequence, *find all type flaws*, and *find all attacks* as the search pruning parameter. For both sequences, the specified claims were verified, and no possible attacks were detected by the tool. The corresponding SPDL scripts are listed in Appendix A.2, and the verification results of parts A, B, C & D, and E & are depicted in Fig. 5.10. This overall verification guarantees that the proposed protocol satisfies the requirements of all timing and nonce's being Alive/ Fresh, weak-agree, Nisynch, and sensitive parameters being secret.

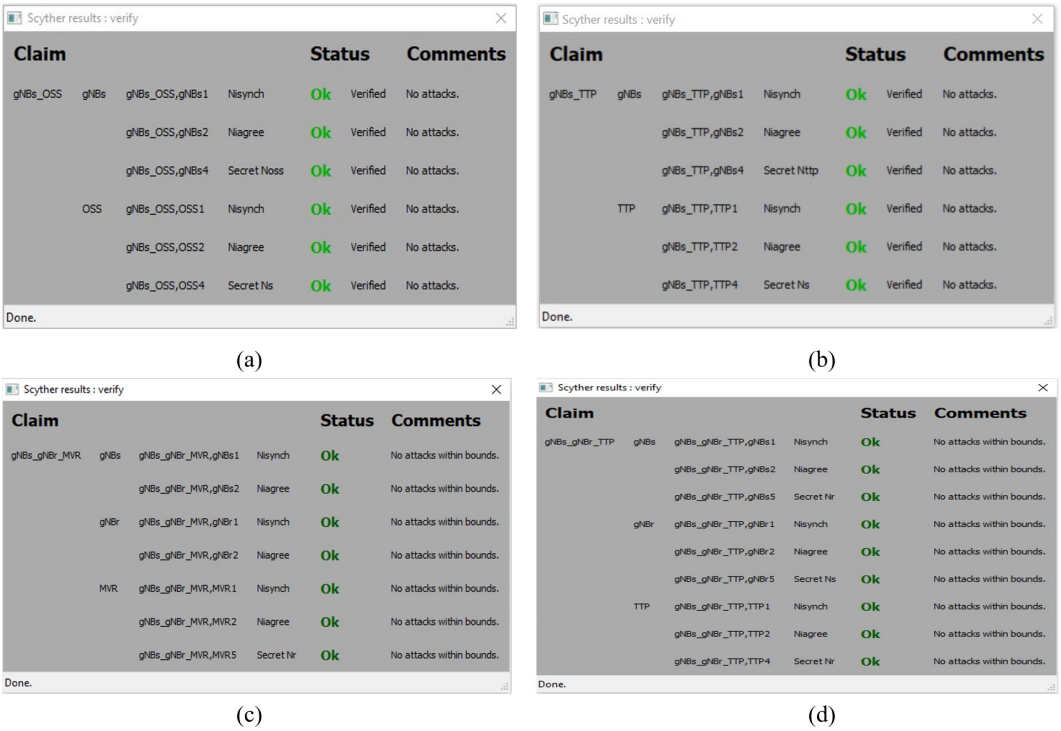


Figure 5.10: Scyther Verification Results of the Protocol: (a) Part A; (b) Part B; (c) Part C & D; (d) Part E & F

5.5.2 Formal verification using the AVISPA

In order to confirm that proposed protocols (i.e., Part A, Part B, and Part C&D) are resilient against the attack, the AVISPA tool [169] is used. AVISPA stands for Automated Validation of Internet Security Protocols and Applications. This tool offers modular and expressive formal language to specify the security features of the authentication protocols. Apart from that, it uses different types of backend server that helps carry out different types of implementation using various automatic analysis techniques ranging from protocol falsification to abstraction-based verification methods for both finite and infinite numbers of sessions. This tool uses the role-based language High-Level Protocols Specification Language (HLPSL) to model the authentication protocol in order to examine its security properties. There are four types of backend servers specified by the tool:1) On-the-fly Model-Checker (OFMC), 2) Constraint Logic-based Attack Searcher (CL-AtSe), 3) SAT-based Model-Checker (SATMC), 4) Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP).

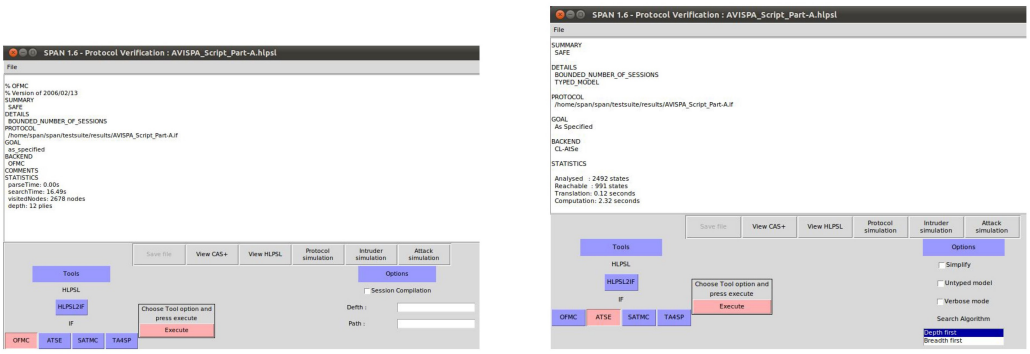
The OFMC and CL-Atse backend servers are used to verify the proposed protocols similar to [155, 170].

5.5.2.1 Simulation of Part A& Part B using AVISPA tool

Part A protocol is modeled into the two roles gNB_{Source} and OSS, sessions, and environment in order to carry out the simulation. The same was used for Part B, two roles gNB_{Source} and TTP, session, and environment() to carry out the simulation. Fig 5.11-(a), Fig 5.11-(b), Fig 5.12-(a), and Fig 5.12-(b) show that the protocol is safe and secure.

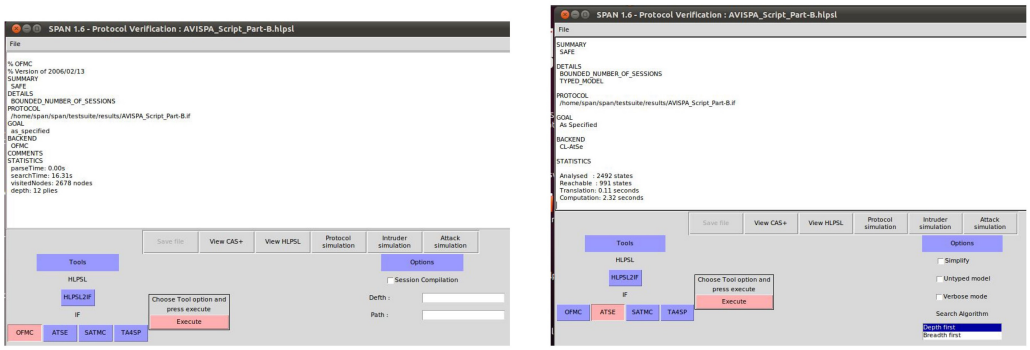
5.5.2.2 Simulation of Part C&D using AVISPA tool

In Part, C&D, protocol is modeled into the three roles gNB_{Source}, $gNB_{Roaming}$, and OSS, sessions, and environment in order to carry out the simulation. Fig 5.13-(a), and Fig 5.13-(b) show that the protocol is safe and secure. The corresponding HLPSL scripts are presented in Appendix A.3.



(a) (b)

Figure 5.11: AVISPA outcome for Part A using (a) OFMC backend server (b) CL-Atse backend server



(a) (b)

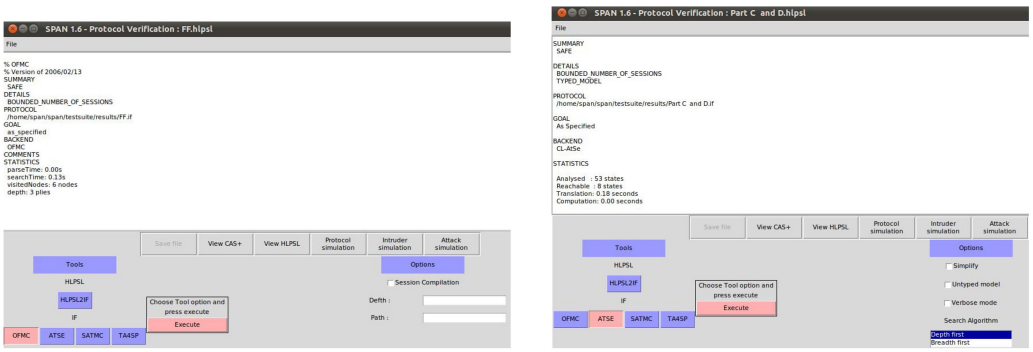
Figure 5.12: AVISPA outcome for Part B using (a) OFMC backend server (b) CL-Atse backend server

5.5.3 Formal security analysis using GNY logic

This section discusses the formal verification of the proposed protocol using the GNY logic [171] (i.e., extended version of BAN logic) to prove the robustness of the proposed protocol and securely mutually authenticate each other.

5.5.3.1 GNY Notations

Let A and B be the two entities communicating, and m be the message. We have used the general symbolism described in [171] to specify the GNY logic, while the logical postulates of Being told rules, Possession rules, and Fresh-



(a) (b)

Figure 5.13: AVISPA outcome for Part C using (a) OFMC backend server (b) CL-Atse backend server

ness rules were utilized in the proving process.

Table 5.3: GNY Notations

Symbols	Description
$A \ni M$	A possess the M
$A \triangleleft *M$	A receives M, which he has not previously communicated in this session.
$A \triangleleft M$	A is told formula M, A receives the M.
$A \mid\sim M$	A once receives the M
$A \mid\equiv \phi(M)$	A assumed M to be recognised.
$A \mid\equiv \#(M)$	A believe that M is new and has never been utilised before.
$\{M\}_K$	K is a shared secret key that encrypts the message M.
$A \mid\equiv A \overset{K}{\leftrightarrow} B$	A thinks that A and B share the secret key (K).
$\overset{K^+}{\longmapsto} A$	K^+ is public key of A, The private key (K^-) will never be obtained by anyone except A.

5.5.3.2 Logical postulates

This section contains the rules that are used to prove the protocol.

Being Told Rule (BTR_1) : If A gets M and has not previously communicated

it in this session, A receives M.

$$\sum_{i=1}^{n} \frac{(1+i)^n}{cf_n(1+i)^n}, n_i \frac{A \triangleleft *M}{A \triangleleft M}$$

BTR_2: If C receives concatenateA message L, M then C coulA also receives the L

$$\sum_{i=1}^{n} \frac{(1+i)^n}{cf_n(1+i)^n}, n_i \frac{C \triangleleft (L, M)}{C \triangleleft L}$$

Being Told rule (BTR_2): If A is given hashed form and one of the two arguments (M, N), the other argument is considered to have been given as well. Where F is a one-to-one function that may be computed in its inverse. If A gets M and hasn't yet communicated it, A accepts N.

$$\frac{A \triangleleft F(M, N), A \ni M}{A \triangleleft N}$$

(2) Being Told Rule (BTR_2) : If A gets a message M that is encrypted using K and A has K, A will be able to receive and decode the M.

$$\sum_{i=1}^{n} \frac{(1+i)^n}{cf_n(1+i)^n}, n_i \frac{A \triangleleft \{M\}_K, A \ni K}{A \triangleleft M}$$

Being Told Rule (BTR_3) : Only A can decrypt the M if it sees the M encrypted with K^+ and has the K^-.

$$\sum_{i=1}^{n} \frac{(1+i)^n}{cf_n(1+i)^n}, n_i \frac{B \triangleleft \{M\}_{k^+}, B \ni K^-}{B \triangleleft M}$$

Being Told Rule (BTR_4) : If A gets a message M that is encrypted using K and A has K, A will be able to receive and decode the M.

$$\sum_{i=1}^{n} \frac{(1+i)^n}{cf_n(1+i)^n}, n_i \frac{A \triangleleft \{M\}_K, A \ni K}{A \triangleleft M}$$

Possession rule (PR_1) : If A gets M, A is presumed to have M.

$$\sum_{i=1}^{n} \frac{(1+i)^n}{cf_n(1+i)^n}, n_i \frac{A \triangleleft M}{A \ni M}$$

PR_2: If C possess L then he can possess the one-way hash function on that message

$$\sum_{i=1}^{n} \frac{(1+i)^n}{cf_n(1+i)^n}, n_i \frac{C \ni (L, M)}{C \ni L}$$

Possession (PR_2) rule : A may be presumed to have $H(M)$ if it has M.

$$\frac{A \ni M}{A \ni H(M)}$$

Possession rule (PR_3): If A has (M, N), it may be inferred that A has (N).

$$\frac{A \ni M, N,}{A \ni (N)}$$

Freshness rule (FR) :If A thinks M is new, then A has the right to believe that any message containing M is fresh, as well as a computationally viable one-to-one function of the message contents.

$$\sum_{i=1}^{n} \frac{(1+i)^n}{cf_n(1+i)^n}, n_i \frac{A \mid\equiv \#(M)}{A \mid\equiv \#(M, N), A \mid\equiv \#(F(M, N)),}$$

FR_2: If C believes that L is fresh and C possess K then C believes that encryption and decryption done by fresh keys K.

$$\sum_{i=1}^{n} \frac{(1+i)^n}{cf_n(1+i)^n}, n_i \frac{C \mid\equiv \#(L), C \ni K}{C \mid\equiv \#(\{L\}_k), C \mid\equiv \#(\{L^-\}_k),}$$

(6) Recognizability rule (RR) : If A thinks that M is identifiable, then A has the right to believe that any message containing M is recognisable as well.

$$\sum_{i=1}^{n} \frac{(1+i)^n}{cf_n(1+i)^n}, n_i \frac{A \mid\equiv \phi(M)}{A \mid\equiv \phi(M, N), A \mid\equiv \phi F(M, N),}$$

(7) Message Interpretation rule (MIR_1) :If A thinks that M is encrypted with K, A owns K, A believes K is shared between B & A, A believes M is recognised, and A believes K&M are new, then A is entitled to believe that B sent the M, B owns K, and A believes that B transmitted the encrypted message.

$$\sum_{i=1}^{n} \frac{(1+i)^n}{cf_n(1+i)^n}, n_i \frac{A \triangleleft \{M\}_K, A \ni K, A \mid\equiv A \overset{K}{\leftrightarrow} B, A \mid\equiv \phi M, A \mid\equiv \#(M,K)}{A \mid\equiv B \mid\sim M, A \mid\equiv B \ni K, A \mid\equiv B \mid\sim \{M\}_K}$$

(8) Message Interpretation rule (MIR_2): If A thinks that M is encrypted with K^+, A has K^-, A believes K^+ , A believes that M is recognizable and A believes M is fresh then A is entitled to believe B has sent the L, A posses K^- and A believes that B sent the encrypted message.

$$\sum_{i=1}^{n} \frac{(1+i)^n}{cf_n(1+i)^n}, n_i \frac{A \triangleleft \{M\}_{K^+}, A \ni K^-, A \overset{K^+}{\longmapsto} A, A \mid\equiv \phi M, A \mid\equiv \#(M)}{A \mid\equiv B \mid\sim M, A \mid\equiv B \ni K^+, A \mid\equiv B \mid\sim \{M\}_{K^+}}$$

(9) Message Interpretation rule (MIR_3): If A thinks that B has sent the M and M is fresh then A entitled to believes that B possess the M.

$$\sum_{i=1}^{n} \frac{(1+i)^n}{cf_n(1+i)^n}, n_i \frac{A \mid\equiv B \mid\sim M, A \mid\equiv \#(M)}{A \mid\equiv B \ni M}$$

5.5.3.3 Security verification of Part A of the proposed protocol that takes place between the gNB_s and the OSS using the GNY logic

Initial assumptions for the protocol

$H_1 : gNB_S \ni PrK_s, H_2 : gNB_S \ni n_s1, H_3 : gNB_S \ni SyK_A$

$H_4 : gNB_S\#(TS2, TS4), H_5 : gNB_S \ni (ID_S, ID_{OSS})$

$H_6 : OSS \ni PrK_{OSS}, H_7 : OSS \ni ID_{OSS}, ID_S$

$H_8 : OSS\#(TS1, TS3), H_9 : OSS \ni n_{OSS}$

The protocol's security goals are as follows:

$OSS \ni H(ID_S, ID_{OSS}, TS_1)$

$gNB_S \ni H(K_{dos}, n_{OSS}, ID_S, TS2, ID_{OSS})$

$gNB_S \ni (SOCK_{TTP}, Cert_{TTP})$

The idealized form of the proposed protocol:

M_{10}: $OSS\triangleleft : *(*ID_S, *TS_1, P_A1)_{PuK_{OSS}}$,

M_{11}: $OSS\triangleleft$: $*H(*ID_S, *ID_{oSS}, *TS_1)$,

M_{20}:$gNB_S\triangleleft$:$*(*k_{dos}, *n_{OSS}, P_A2, *TS_2, R2)_{PuK_S}$

M_{21}:$gNB_S\triangleleft$:$*H(*k_{dos}, *n_{OSS}, *ID_S, *TS2*)$

M_{30}:$OSS\triangleleft$:$*(*n_s1, *X, H[n_{OSS}, TS_3], *TS_3])_{PuK_{OSS}}$,

M_{31}:$OSS\triangleleft$: $H(*(*n_{OSS}, *TS_3)$,

M_{40}:$gNB_S\triangleleft$:$*(*(SOCK_{TTP}, *Cert_{TTP})_{SyK_A}, *H[n_s1,$

$TS_4], *TS_4)_{PuK_s}$

M_{41}:$gNB_S\triangleleft$: $*H(*n_s1, *n_{OSS}, *X)$.

Proof and derivation of security goals:

By applying the BTR_1, BTR_3 and Pr_1 rule on M_{10} based on H_7, we get

$S_1 : S_1 : OSS \ni (ID_S, TS_1, P_A1)$

We apply the BTR_2 and PR_2 rule based on S_1 and H_7, we get

$S_2 : OSS \ni H(ID_S, ID_{OSS}, TS_1)$

Applying the FR rule on S_1 and S_2 based on S_1 and H_8 we get

$S_3 : OSS \mid\equiv \#(ID_S, ID_{OSS}, P_A1)$

By applying the BTR_1, BTR_3 and Pr_1 rule based on H_1, we get

$S_4 : gNB_S \ni (K_{dos}, n_{OSS}, TS2, P_A2)$

We apply the BTR_2 and PR_2 rule based on H_5, S_4

$S_5 : gNB_S \ni H(K_{dos}, n_{OSS}, ID_S, TS2, ID_{OSS})$

Applying the FR rule based on S_4 and H_4, we get

$S_6 : OSS \mid\equiv \#(K_{dos}, n_{OSS}, ID_S, P_A2, ID_{OSS})$

By applying the BTR_1, BTR_3 and PR_1 rule on M_{30} based on H_7, we get

$S_7 : OSS \ni (n_s1, X, H[n_{OSS}, TS3], TS3])$

We apply the BTR_2 and PR_2 rule on M_{31} based on, H_9 and S_7

$S_8 : OSS \ni H(n_{OSS}, TS3)$

Applying the FR rule based on S_8 and H_8, we get

$S_9 : OSS \mid\equiv \#(n_s1, X, H[n_{OSS}])$

By applying the BTR_1, BTR_3 and Pr_1 rule on M_{40} based on H_1, we get

$S_{10} : gNB_S \ni ((SOCK_{TTP}, Cert_{TTP})_{SyK_A}, H[n_s1, TS_4], TS_4)$

We apply the PR_3 rule on S_{10}

$S_{11} : gNB_S \ni (SOCK_{TTP}, Cert_{TTP})_{SyK_A}$

We apply the BTR_4 rule on S_{11} based on $H3$

$S_{12} : gNB_S \ni (SOCK_{TTP}, Cert_{TTP})$

We apply the BTR_2, PR_3 rule on M_{41} based on S_{10}, H_{12}

$S_{13} : gNB_s \ni H(n_s1, n_{OSS}, X)$

Applying the FR rule based on on S_{10} and H_3, we get

$S_{14} : gNB_s \mid\equiv \#(SOCK_{TTP}, Cert_{TTP}, H[n_s1, TS_4]))$

Since Part B of the protocol is identical to Part A, we can prove that Part B follows the same approach.

5.5.3.4 Security verification of Part C and D of the proposed protocol that takes place between the gNB_s, gNB_R and the TTP using the GNY logic

Initial assumptions for the protocol

$H_1 : gNB_S \ni PrK_s, H_2 : gNB_S \ni ID_S, ID_{TTP}, ID_{MP}$

$H_3 : gNB_S \#(TS4, TS4), H_4 : gNB_S \ni (SyK_{C1}, SyK_{C2})$

$H_5 : gNB_R \#(TS1, TS3), H_6 : gNB_R \ni PrK_R$

$H_7 : gNB_R \ni n_R1, H_8 : gNB_R \ni (ID_R, ID_{TTP}, ID_{MP})$

$H_9 : TTP \ni (ID_R, ID_{TTP}, ID_{MP}), H_{10} : TTP \#(TS2)$

$H_{11} : TTP \ni PrK_{TTP}, H_{12} : gNB_S \ni (SyK_D)$

The protocol's security goals are as follows:

$gNB_R \ni TTP_{ID}, MP_{ID(R0)}, SOCK_{TTP_M}$,

$Cert_{TTP}, n_s3, TS1, gNB_R$

$gNB_R \ni (H((n_R1, TS_3))$

$gNB_s \mid\equiv OSS \ni (SOCK_{TTP}, Cert_{TTP})$

$gNB_s \mid\equiv OSS \mid\equiv U_E \overset{SK}{\longleftrightarrow} H_N$

The following steps demonstrate the idealized form of the proposed protocol:

$M_{10} : gNB_R \triangleleft : *(*ID_S, *ID_{TTP}, *ID_{MP}, *SOCK_{TTP},$

$*Cert_{TTP}, *n_s3, P_C1, *TS1)_{PuK_R}$,

M_{11}: $gNB_R \triangleleft$: $H(*ID_S, *ID_{TTP}, ID_{MP}, *SOCK_{TTP},$

$Cert_{TTP} * n_s3, ID_R, *TS_1)$,

$M_{20} : TTP \triangleleft$: $*(*ID_R, *ID_{TTP}, ID_{MP}, n_R1, P_C1, P_C2,$

$TS2)_{PuK_{TTP}}$

M_{21}:$TTP \triangleleft$: $*H(*ID_R, *ID_{TTP}, *ID_{MP}, n_R1, TS2))$

$M_{30} : gNB_R \triangleleft : *(ID_{MP}, P_D2, (CODE_M, r_1, r_2, N)_{SyK_D},$

$H[n_R1, TS3], TS3)_{PuK_R}$,

$M_{31} : gNB_R \triangleleft$:$H(*(*n_R1, *TS_3)))$,

$M_{40} : gNB_S \triangleleft$:$*(*ID_{TTP}, *ID_{MP}, P_C2, (CODE_M, r_2')_{SyK_{C1}},$

$*TS4)_{PuK_s}$,

$M_{50} : gNB_S \triangleleft$: $*(*ID_R, *P_C3, *(CODE_M, n_R2)_{SyK_{C2}}, *n_R2,$

$H[n_R3, TS5], TS5)_{PuK_S}$,

$M_{51} : gNB_s \triangleleft$: $*H(*n_S3, TS5)$,

Proof and derivation of security goals:

By applying the BTR_1, BTR_3 and PR_1 rule on M_{10} based on H_6, we get

$S_1 : gNB_R \ni (ID_S, ID_{TTP}, ID_{MP}, SOCK_{TTP},$
$Cert_{TTP}, n_s3, P_C1, TS1)$

BTR_2 and PR_2 rules on M_{10} based on S_1 and H_8 was applied.

$S_2 : gNB_R \ni H(ID_S, ID_{TTP}, ID_{MP}, SOCK_{TTP},$
$Cert_{TTP}, n_s3, ID_R, TS_1)$

Applying the FR rule on S_1, S_2 based on H_5, we get

$S_3 : gNB_R \mid\equiv \#(ID_S, ID_{TTP}, ID_{MP}, SOCK_{TTP},)$
$Cert_{TTP}, n_s3, P_C1$

By applying the BTR_1, BTR_3 and PR_1 rule on M_{20} based on H_{11} , we get

$S_4 : TTP \ni (ID_R, ID_{TTP}, ID_{MP}, n_R1, P_C1, P_D1, TS2)$

We apply the BTR_2 and PR_2 rule on M_{21} based on S_4 and H_9

$S_5 : TTP \ni H(ID_R, ID_{TTP}, ID_{MP}, n_R1, TS2)$

Applying the FR rule on S_4 based on H_{10}, we get

$S_6 : TTP \mid\equiv \#(H(ID_R, ID_{TTP}, ID_{MP}, n_R1, P_C1, P_D1,))$

By applying the BTR_1 and BTR_3 and PR_1 rule on M_{30} based on H_6, we get

$S_7 : gNB_R \ni (ID_{MP}, P_D2, (CODE_M, r_1, r_2, N)_{SyK_D},$
$H[n_R1, TS3], TS3)$

We apply the BTR_2 and PR_2 rule on M_{31} based on, H_8 and S_6

$S_8 : gNB_R \ni (H(n_R1, TS_3))$

We apply the PR_3 rule on S_7

$S_9 : gNB_R \ni (CODE_M, r_1, r_2, N)_{SyK_D}$

We apply the BTR_4 rule on S_9 based on H_{12}

$S_{10} : gNB_R \ni (CODE_M, r_1, r_2, N)$

Applying the FR rule based on S_5 and H_5 we get

$S_{11} : gNB_R \mid\equiv \#(ID_{MP}, P_D2, (CODE_M, r_1, r_2, N),)$
$H[n_R1, TS3], TS3$

By applying the BTR_1, BTR_3 and PR_1 rule on M_{40} based on H_1, we get

$S_{12} : gNB_s \ni ID_{TTP}, ID_{MP}, P_c2, (CODE_M, r_2')_{SyK_{C1}}, TS_4$

We apply the PR_3 rule on S_{12}

$S_{13} : gNB_S \ni (CODE_M, r_2')_{SyK_{C1}}$

We apply the BTR_4 rule on S_{13} based on H_4

$S_{14} : gNB_S \ni (CODE_M, r_2')$
Applying the FR rule based on S_{10} and H_3, we get
$S_{15} : OSS \mid\equiv \#(ID_{TTP}, *ID_{MP}, P_C2, (CODE_M, r_2')),$
$*TS_4$
By applying the BTR_1, BTR_3 and PR_1 rule on M_{50} based on H_1, we get
$S_{16} : gNB_s \ni ID_R, P_C3, (CODE_M, n_R2)_{SyK_{C2}}, n_R2,$
$H[n_R3, TS5], TS5$
We apply the BTR_2 and PR_3 rule on M_{51} based on S_{16}
$S_{17} : gNB_S \ni H(n_S3, TS5)$
We apply the PR_3 rule on S_{15}
$S_{18} : gNB_S \ni (CODE_M, n_R2)_{SyK_{C2}}$
We apply the BTR_4 rule on S_{13} based on H_4
$S_{19} : gNB_S \ni (CODE_M, n_R2)$
Applying the FR rule based on S_{16} and H_3, we get
$S_{20} : OSS \mid\equiv \#(ID_R, M2, (CODE_M, n_R2), n_R2,)$
$H[n_R3, TS5]$

Since Parts E & F of the protocol is identical to Parts C & D, we can prove Parts E & F in the same manner.

5.5.4 Formal security analysis using ROR Logic

We use Real-or-Random (ROR) logic proposed by Abdalla et al. [172] in order to verify the session key security. ROR is considered a more suitable case than the original model proposed by Bellare et al. [173] to formally simulate real attacks on the authentication scheme for 5G gNodeBs in Service Migration Scenarios of MEC. Some of the basic concepts following their work are omitted in this thesis, such as participants, long-term keys, freshness, etc. In the ROR model, the security model is defined by a game between two probabilistic polynomial-time Turing machines, namely a challenger (CH) and an adversary (A_d). It is assumed that CH belongs to a real system that has already applied our proposed scheme P_{MEC} to it. In order to evaluate P'_{MEC} security, CH intended to invite A_d to launch a real attack on P_{MEC}, but CH worried about A_d would learn enormous useful information about the real sys-

tem. So CH developed an oracle system and designed a game to play with A_d. After initialization, the oracle flips an unbiased coin c (c=0, 1), and the goal of A_d is to guess the value of CH. To increase the chance of winning this game, A_d is provided a series of queries to ask the oracle. There are three communicating parties, such as (gNB_{Sourec}, $gNB_{Roaming}$, TTP) involved in the authentication of 5G gNodeBs in Service Migration Scenarios of MEC. Let instances of f, g and h of gNB_{Sourec} , $gNB_{Roaming}$ and TTP are denoted by gNB_S^f and gNB_R^g and TTP^h respectively. In the ROR model, it is assumed that A_d can perform several activities such as delete, insert and edit the captured exchanged message. A_d can perform these activities by executing the queries defined in the ROR model. The description of these queries is as follows.

- *Execute (gNB_S^f , gNB_R^g, TTP^h):* A_d executes this query to intercept the exchanged message between the gNB_S^f , gNB_R^g, TTP^h.
- *Reveal* $((\Pi^l)$*:* A_d executes this query to obtain the current session key between the gNB_S^f, gNB_R^g, TTP^h.
- *Send* $((\Pi^l, mess)$*:* A_d executes this query to forge the captured message (i.e., it modified the captured message and then replay this message to the gNB_S^f, gNB_R^g, TTP^h.) so that he receives the response of the forged message.
- *Test* (E^h)*:* This query is used to examine the session key security of the communicating entities D^f, and $AS^g)$. In order to examine the session key security, a coin is tossed before starting the game. Based on the tossed outcome, A takes a decision (i.e., c=0, communicating party returns the random number or c=1, then communicating party returns the session key. Otherwise, a null value is returned.)

Theorem 1: If A_d tries to crack the session key (SK) in polynomial time. Then $Adv_{A_d} \leq \frac{H_Q^2}{2^U} + 2A_{A_d}^{ECDDH}$

Where H_Q, A^{ECDDH}, U stands for a number of $Hash$ queries, hardness of the discrete logarithm problem, and hash function output value, respectively.

Proof: Since Part A, Part B, Part C&D, and Part E&F use the combination of ECC and RSA to protect the exchange message confidentiality and integrity. Here, we do the proof for Part C&D since all the parts employ the same mechanism. The proof of the protocol is shown using the three games known as G_1, G_2, G_3. An event $S_{A_dG_1}$ is defined as the success probability of the A_d to guess the session key or to win the game.

Game (G_1)**:** By executing this game, A_d tries to get the actual value of c at the start of the game before Oracle initializes the procedure of the game. So,

$$Adv_{A_d} = |2P[S_{A_dG_0}] - 1] \tag{5.1}$$

Game (G_2)**:** A_d executes the *Execute* query to win the game by intercepting the exchange message $\{M_C1, M_C2, M_C3, M_D1, M_D2\}$ between the $gNB_{SOURCE}, gNB_{Roaming}$, and TTP. When A_d obtains the exchange message, then it tries to get the correct secret value by guessing the value of c based on the execution of *Test* query. Since we use the random numbers $\{a_3, b_3, c, a_4, a_5\}$ derived from the elliptic curve. They are random (i.e., used only once in the protocol) and exchanged in encrypted form between the communicating entities. So, finding any clue for the random numbers is tough so that A_d will get the session key. Therefore, A_d will lose the game, and the winning possibility of G_1 will be similar to the G_2. Hence we can get

$$P[S_{A_dG_2}] = P[S_{A_dG_1}] \tag{5.2}$$

Game(G_3)**:** Since during the G_2 execution attacker was unable to get the right session key, but he has the intercepted message. In this game, A_d tries different ways to get the session key by modeling this game as an active attack by executing the *Send* query. We use the ECC and RSA that protects the exchange message and will not let the A_d derive any secrets of the protocol, especially to determine the random number from these computed parameters $\{P_c1, P_C2, P_C3P_D1, P_D2\}$ due to the hardness of the discrete logarithm problem [166]. This shows that A_d will not be able to determine any insight to derive the session key and will not be any collision while the *Hash* will run. So,

the winning probability of G_3 will be similar to the previous game. Hence, This can be obtained by adopting the birthday paradox

$$P[S_{A_d G_2}] - P[S_{A_d G_3}] \leq \frac{H_Q^2}{2^{U+1}} + +2A_{A_d}^{ECDDH} \tag{5.3}$$

Now, all the game has been executed by the A_d in order to predict the correct value of C, So we can

$$P[S_{A_d G_3}] = \frac{1}{2} \tag{5.4}$$

from Eq(5.1) (5.2), and (5.4), we can obtain

$$\begin{aligned} Adv_{A_d} &= |2P[S_{A_d G_1}] - 1| \\ \frac{1}{2}Adv_{A_d} &= |P[S_{A_d Game_1}] - \frac{1}{2}| \\ &= P[S_{A_d G_2}] - P[S_{A_d G_3}] \end{aligned} \tag{5.5}$$

We obtain the following outcome from Eq (5.3) and (5.5).

$$\begin{aligned} \frac{1}{2}Adv_{A_d} &\leq \frac{H_Q^2}{2^{U+1}} + 2A_{A_d}^{ECDDH} \\ Adv_{A_d} &\leq \frac{H_Q^2}{2^U} + 2A_{A_d}^{ECDDH} \end{aligned} \tag{5.6}$$

The outcome after executing the game indicates that A_d can not obtain the session key in a polynomial amount of time.

5.6 Validation and Performance Comparison

This section addresses the proposed protocols' performance measurements in terms of security verification, computational, communication, and energy consumption and compares them to their equivalent counterparts in the literature.

5.6.1 Security features verification

This section presents the security features verification as mentioned in Section 5.2.3. The conducted informal and formal (Scyther and GNY logic) security verification shows the proposed protocol's robustness against the identified attacks. The research carried out in [174, 175] shows that [155] is vulnerable to various attacks such as Anonymity and MitM. [155, 176] shows that [177] does not offer the perfect forward secrecy, no formal verification, and Anonymity. The security analysis depicted in [178, 155] confirms that[170] does not offer Anonymity and is also vulnerable to traceability attack. Further, [179] does not offer anonymity, and is vulnerable to DoS threats according to [180, 155, 181].

The comparison outcome shown in Table 5.4 demonstrates that the proposed protocol has the capability to offer all the identified security features while [155, 177, 170, 179] fails in some sort of security features verification. The protocol in [182], however, meets all the security features covered in our protocol. Despite its merits in terms of the security features coverage, the [182] targets a handover authentication between the gNB and the UE, unlike in MEC-SMAP, where the focus is on SMC. The fact that we employ a random number, timestamp, and nonce that changes after each successful authentication is the major rationale for providing all of the security features.

Table 5.4: Comparing security features of existing protocols/L_1-Mutual Authentication; L_2-Anonymity; L_3-PFS; L_4-Replay protection; L_5-DoS protection; L_6-Traceability protection; L_7-Protection from malicious gNBs; L_8-Formal analysis

Protocols	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8
[155]	✓	×	✓	✓	✓	✓	✓	✓
[177]	✓	×	×	✓	✓	✓	✓	×
[170]	✓	×	✓	✓	✓	×	✓	✓
[182]	✓	✓	✓	✓	✓	✓	✓	✓
[179]	✓	×	✓	✓	×	×	×	×
Ours	✓	✓	✓	✓	✓	✓	✓	✓

5.6.2 Computational Cost

This section evaluates the number of cryptographic operations used in the proposed protocol and its counterparts. We consider the cost of cryptographic

operation as mentioned in [155], using two cores on Intel i7-6600U CPU @ 2.60 GHz as $gNB's$ and the OpenSSL with two cores on Intel i4-2500 @3.30 GHz as OSS shown in Table 5.5. The notation T_{RSA}, T_H, T_P, T_E, T_{SM} and T_{MSM} stands for RSA signature, Secure Hash Algorithm ($SHA2$) function, pairing operation, modular exponential, elliptic curve scalar multiplication and multi elliptic curve scalar multiplication, respectively. Table 5.6 contains the computational cost for all the proposed protocols. We also compare the proposed protocol (Part A) with [155][177][170][182] [179] in terms of computational cost, which is displayed in Table 5.8, demonstrating that the proposed protocol is the least expensive. The proposed protocol is the least costly since it is based on the combination of T_{RSA} and T_{SM} which is less costly compared to [155][177][170][182] [179] that uses the T_E, T_{SM} and T_{MSM} as shown in Table 5.5 obtained through experimental analysis as [155]. As stated earlier, current literature lacks any authentication mechanisms to contrast against phase C& D and phase E& F.

Table 5.5: Computation cost for cryptographic operations

Protocols	T_{SM}	T_H	T_{MSM}	T_{RSA}	T_P	T_E
gNB	0.2025	0.0032	0.2532	0.127	2.87	0.225
OSS	0.03	0.00029	0.0375	0.019	0.7616	0.0337

5.6.3 Communication Cost

This section evaluates the number of bits transmitted in the channel for the proposed protocol and its counterparts. To compute the number of bits transmitted, we use the same bit size as used in [155]. The notation M_{RSA}, M_{ID}, M_{TS}, M_{SM} and M_H stands for the packet is encrypted with RSA size of 2048 bit, the identity of 32 bits, timestamp of 32 bit, elliptic curve multiplication of 224 bits and hashed by 256 bit, respectively. Table 5.7 shows the communication cost for the proposed protocols. We also compare the proposed protocol (Part A) with [155][177][170][182] [179] in terms of communication cost, which is illustrated in Table 5.8, indicating that the proposed protocol takes the high-cost [155][177][170][182] and is the less compared to [179]. Although our proposed protocol takes communication cost high cost compared

to [155][177][170][182], but offers the better security, less computational cost, and additional parts such as part C&D and part E& F.

5.6.4 Storage Cost

This section determines the memory required for the gNB_s and gNB_R. We take the size of cryptographic operations as stated in [155], with RSA being 2048 bits, identity being 32 bits, and hashed output being 256 bits. Table 5.7 shows the storage cost required for the proposed protocols.

5.6.5 Energy Consumption

This section determines the energy consumption for the cryptographic operations utilized in the various part of the proposed protocol. We have used the same energy consumption as [183] to measure energy consumption. To compute the energy consumption, experiments are performed using a "Strong ARM" CPU running at 133 MHz doing various tasks is described as the energy required for transmitting a bit, AES symmetric enc/dec, Hashed output, enc/dec RSA are 0.00066 mj, 0.00217 mj, 0.000108 mj, 15.3 mj, respectively. Table 5.6 shows the energy consumption required for the proposed protocols.

Table 5.6: Computational cost and energy consumption of the proposed protocols

Protocol Part	Computational cost	Time (ms)	Energy required	Energy consum. (mj)
Part A	$6T_{RSA}$+$6T_H$+$4T_{SM}$+$1T_{AES}$	0.91	(12800×0.00066+6×0.000108+6×15.1+4×8.8+1×.00208)	134.24
Part B	$6T_{RSA}$+$6T_H$+$4T_{SM}$+$1T_{AES}$	0.91	(12800×0.00066+6×0.000108+6×15.1+4×8.8+1×.00208)	134.24
Part C&D	$10T_{RSA}$ + $10T_H$ + $9T_{SM}$ + $3T_{AES}$	2.14	(18994×0.00066+10×0.000108+10×15.1+9×8.8+3×.00208)	242.74
Part E&F	$7T_{RSA}$+$8T_H$+$6T_{SM}$+$3T_{AES}$	1.97	(14848×0.00066+8×0.000108+7×15.1+6×8.8+3×.00208)	168.31

Table 5.7: Communication and storage costs of the proposed protocols

Protocol Segment	Message exchanges	Communication cost (bits)	Stored credentials	Storage cost (bits)
Part A	$((2M_{RSA} + M_H), (2M_{RSA} + M_H), (M_{RSA}), (M_{RSA}))$	12800	$(ID_{OSS}, ID_S, PrK_{OSS}, PrK_S)$	4160
Part B	$((2M_{RSA} + M_H), (2M_{RSA} + M_H), (M_{RSA}), (M_{RSA}))$	12800	$(ID_{TTP}, ID_S, PrK_{TTP}, PrK_S)$	4160
Part C& D	$((2M_{RSA} + M_H), (2M_{RSA} + M_H), (2M_{RSA}), (M_{RSA}), (2M_{RSA}))$	18994	$(2ID_{TTP}, 2ID_{MP}, 2ID_s, SOCK_{TTP}, Cert_{TTP},$, $2ID_R, PrK_R, PrK_S, PrK_{TTP}, r_1, r_2)$	10560
Part E& F	$(M_{RSA} + M_H), (2M_{RSA} + M_H), (2M_{RSA}), (M_{RSA}), (M_{RSA}))$	14848	$(2ID_{TTP}, 2ID_{MP}, 2ID_S, SOCK_{TTP}, Cert_{TTP},$ $2ID_R, PrK_R, PrK_S, r'_2, PrK_{TTP}, r_1, r_2)$	10592

Table 5.8: Comparison of computational and communication costs of Part A segment with its counterparts

Protocols	UE side	Total time (ms)	Message exchange	Total cost (bits)
[155]	$6T_{SM} + 2T_{MSM}$	0.9982	$((3M_{SM} + M_{ID}, M_{TS}), (3M_{SM} + M_{ID} + M_{TS} + M_H), (M_H))$	1792
[177]	$8T_E + 2T_{RSA}$	1.18	$((2M_{RSA} + M_{ID} + M_{SM} + M_{TS}), (2M_{RSA} + M_{ID} + M_{SM} + M_{TS} + M_H), (M_{TS}))$	9088
[170]	$6T_{SM} + 4T_{MSM}$	1.3		
[182]	$3T_E + 5T_{SM} + 3T_P + T_{MSM}$	3.46	$((M_{RSA} + M_{ID} + 3M_{SM} + M_{TS}), (M_{RSA} + M_{ID} + M_{SM} + M_{TS} + M_H), (M_{TS}))$	5408
[179]	$3T_P + 6T_{SM} + 7T_{MSM}$	10.2	$((M_{ID} + M_{SM} + M_{TS}), (M_{ID} + 3M_{SM} + M_{TS}, (7M_{RSA} + M_{TS}))$	15692
Ours	$6T_{RSA}+6T_H+4T_{SM}+1T_{AES}$	0.91	$((2M_{RSA}+M_H), (2M_{RSA}+M_H), (M_{RSA}), (M_{RSA}))$	12800

5.7 Prototype Implementation

This section explicates the pragmatic feasibility of the protocol while justifying the necessity for the embedded security measures. The specifications of the implemented prototype MEC environment are further elucidated while the conducted experiments are described within the valuation context.

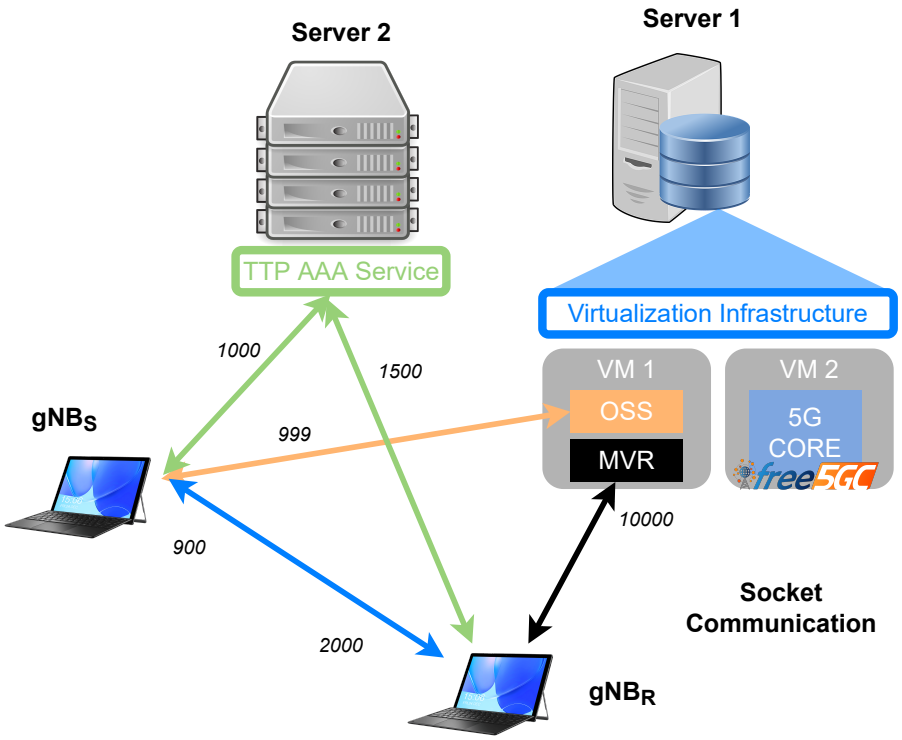


Figure 5.14: Prototype Implementation of the Proposed Protocol

5.7.1 Developed Experimental MEC Environment

A prototype testbed of the MEC service migration environment was developed and emulated for evaluating the feasibility of the proposed protocol in a practical deployment scenario. Fig. 5.14 represents the formation of the entities, gNB_S, gNB_R, TTP, and the OSS. A high-performance server (i.e. Processor: Intel Xeon 2.2 GHz 24 CPU, RAM: 98 GB, OS: Ubuntu 16.04 LTS) was

launched at the MEC system level, that embeds both the OSS and 5G Core entities operating as VMs. The 5G Core was launched leveraging the free5GC (i.e. https://www.free5gc.org/) tool. In addition, the TTP or AAA server function was launched at a separate server bearing moderate specifications of Processor: Intel Xeon 2.4 GHz 4 CPU, RAM: 8 GB, OS: Windows Server 2016 64bit. The MEC virtualization platforms of the two emulating gNBs were maintained in two laptops, due to the requirement for them to be mobile and dynamic to conduct the current and future emulations. The connections between the entities or interfacing were established via the socket-based Inter-Process Communication (IPC) approach. The protocol steps were specified using a Java base. For the cryptographic operations, RSA-4096 bit, AES-256, SHA-512, clock-skew 50 ms, and K_{DoS} as 4 parameters were used. The complexity K_{DoS} exceeding 4 would consume more than 500 ms for solving, which is not ideal for the context of the protocol. The P-256 ECDH construct described in RFC5903 was deployed for relevant ECC-based PFS mechanisms. Further information about the prototype development is available in Appendix A.7. The developed prototype MEC setup converged the protocol to an average completion time of 2047 ms, which covered the phases from A to G. A comparatively higher value is exhibited due to the involvement of many identity and legitimacy verification entities, in addition to the adoption of DoS mitigation and PFS ensuring methods, that incur formidable delays to the protocol.

5.7.2 Conducted Emulation-based Experiments

In order to implicate a scenario where the proposed security measures were not applicable, we have removed the main security measures from the developed protocol and denoted it as the Legacy Protocol (LP). The LP, therefore, withdrew 2 messages from each A and B phases, leaving only the request and reply messages with their inherent Asymmetric security measures. All the timing, hashing, nonce, and DoS measures were detached from the message flows. The details of the LP are available under Appendix A.5. Since we have already conducted a cost-wise comparison with existing protocols in Table 5.8, the impact of integrating the proposed security measures into the

protocol was evaluated to justify their security-heavy nature. Fig. 5.15-(a) presents a comparison between the protocol Completion Times (CTs) of the Standard Protocol (SP) and the LP. With the reduced security overhead, LP converges to an average CT of 814 ms. On the contrary, Fig. 5.15-(b) divulge the impact of a DoS attack on the LP from attempt 21 to 40, where the CTs are clearly accumulated beyond 6000 ms. Fig. 5.15-(c) further elaborates on the impact of a DDoS threat, where DDoS bots were emulated towards the server interfaces of the protocol and assumed that each request was handled sequentially while running the setup. Since there are two server interfaces, C1 represents the case where only the OSS is subjected to the DDoS threat, while C2 represents the scenario where both OSS and TTP entities are under the influence of the stated threat. The emulation deduced that DoS measures are essential to mitigate the wasted timing on the system in addition to the exposed idling server interfaces. Fig. 5.15-(d) depicts a cost-wise perspective of the proposed SP and the LP in case of either tampering or Replay attempts were directed toward the different phases of the protocols. The lack of integrity or freshness measures of LP proves costly.

5.7.3 Simulation to Evaluate the Impact of Tampering

Further, we have simulated the behavior of the protocol in case of a tampering threat, and that is depicted in Fig. 5.16. In this simulation, the protocol completion time was computed considering the delays that ensued for re-transmissions with the perpetrated tampering attempts. This simulation compares the behavior of the two protocols SP and LP, in the context of the probability that a tampering threat is occurring. The developed MatLab code of the simulation is available in Appendix A.6. From Fig. 5.16, it is clearly observable that both SP and LP are converging to their maximum re-transmission delays approximately at 0.2 and 0.3 probabilistic instances. LP protocol is undoubtedly exceeding the delays as it lacks the means to detect the ensued tampering attempts. Thus, security mechanisms integrated into the proposed protocol are vital for safeguarding the entities and content involved with service migration processes of MEC environments.

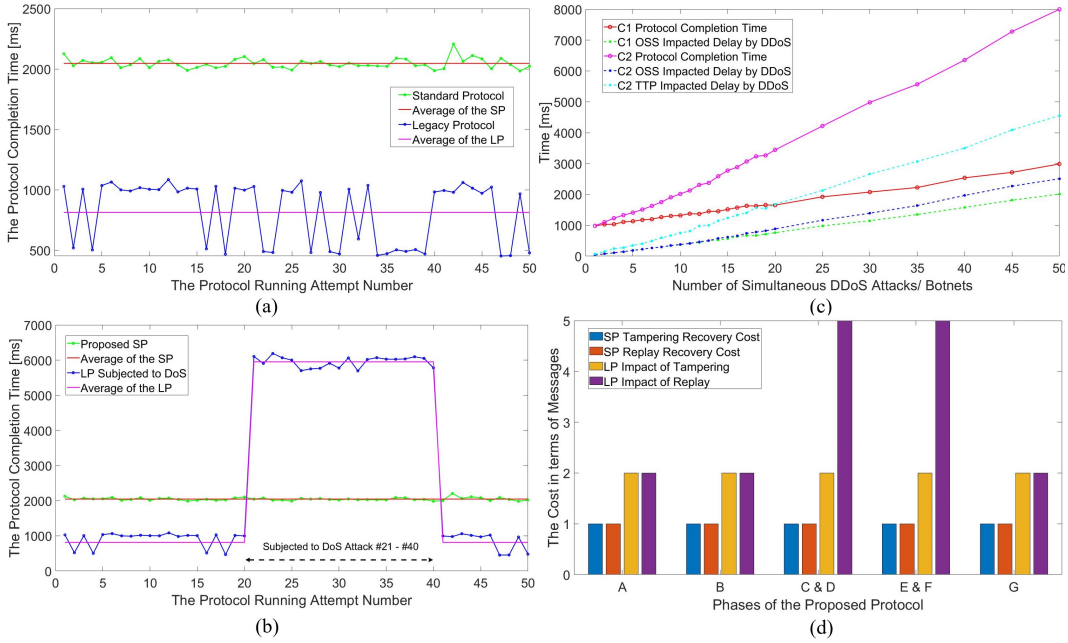


Figure 5.15: The Results of the Emulations Conducted in the Developed Testbed Environment

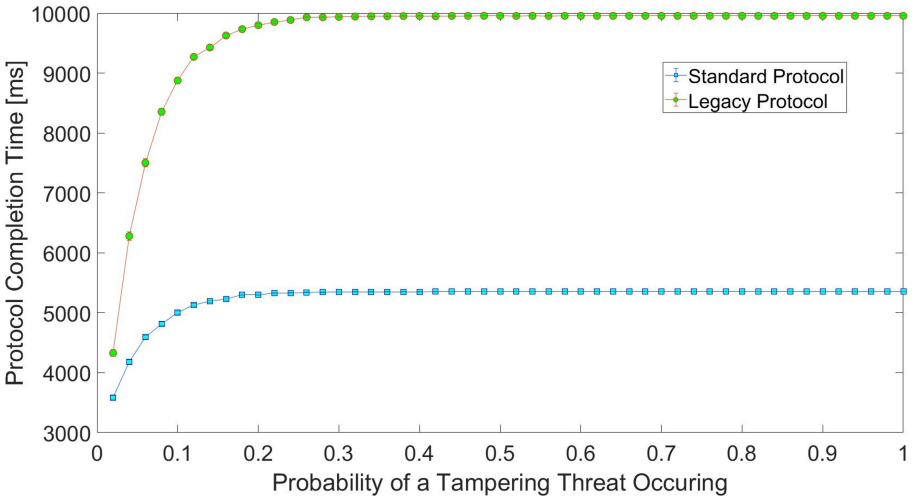


Figure 5.16: The Impact of Tampering to the Protocol Completion Time, based on a Probabilistic Approach

5.8 Chapter Summary and Discussion

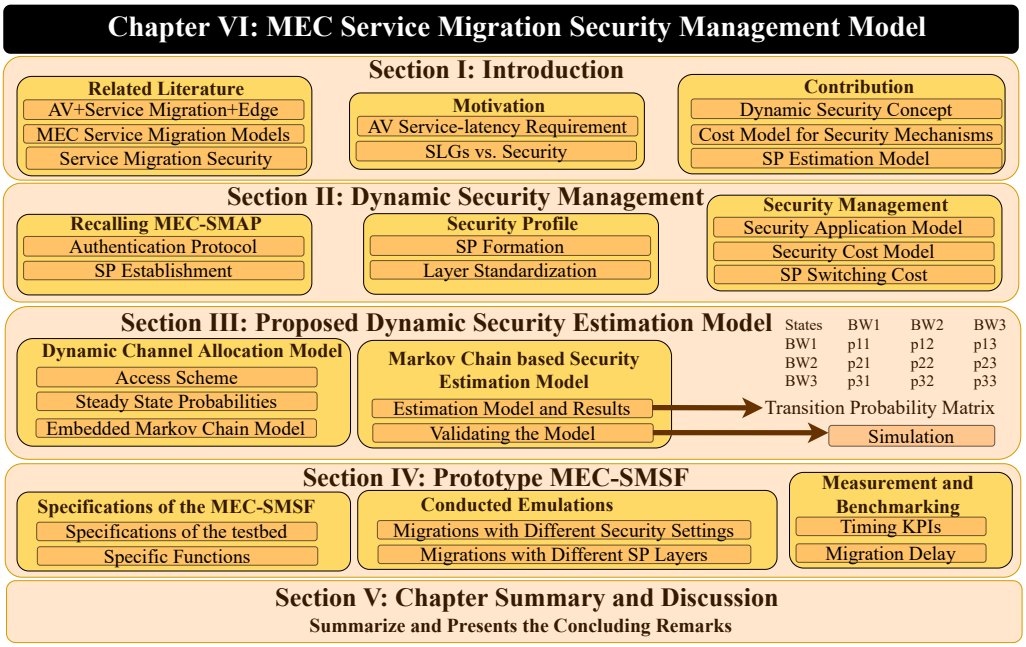
In this chapter, the focus was to address the security issues associated with the edge-to-edge service migration processes of MEC deployments, that take place between two gNBs. As the MEC is deployed in a dynamic environment that feeds the 5G-related services, managing the security in long-lasting migration processes is an obvious conundrum. As the pioneer directive that addressed this predicament, our main focus was to maximize the security measures along with mitigating DoS attempts, and the legitimacy issues that are present due to the decoupling of physical and virtual operating environments in novel MEC-based deployments. The proposed protocol ensures mutual authentication among engaged entities through signature and nonce-based verification methods. In addition, a method for verifying the legitimacy of the operating MESs was proposed. As the protocol embeds service-level eligibility verification schemes, an essential PFS-ensuring method was installed into the message flows. The proposed protocol ensures the formation of the migration master key at the conclusion, while it can be utilized to safely configure the respective security profiles. The conducted formal analysis with Scyther and AVISPA tools along with GNY and ROR logics, and the informal analysis specifications proved the correctness of our protocol. The comparable values of the efficiency measurements and the timing measurements from the testbed implementation prove the feasibility of the protocol in a server environment suited for a MEC deployment.

CHAPTER

SIX

MEC SERVICE MIGRATION SECURITY MANAGEMENT MODEL

Security and Service Level Guarantees (SLGs) are at their divergent ends when it comes to communication technologies; as employed security mechanisms generate security overheads, and latency for encryption/decryption, and result in formidable processing power that alleviates the possibility of meeting the specified SLGs. As service migration processes are restricting the performance of MEC deployments, the security of the migrating content and the latency of the migration channel that contribute to the downfall of SLGs are both imperative parameters. This presents a clear inverse relationship between security and latency that is worth investigating for the feasible deployment of MEC-enabled vehicular services. Thus, this chapter proposes a Service Migration Security Management Model, that specifies the methodology to select the optimal security level for the MEC-SMSF proposed in Section 4.3. The introduced model optimizes the level of security applied for the migrating content based on the instantaneous bandwidth utilization of the channel, that maintains the SLGs. Further, the proposed model and its concepts are validated with simulations and a prototype implementation of the MEC-SMSF.



Chapter Organization

This chapter is formed with 5 sections. Section 6.1 presented the motivation for this work in relation to the use case of AVs explicated in Section 1.1.2. After the related literature, the contributions of this chapter are mentioned in the same section. The concept of Dynamic Security Management is presented in Section 6.2, where the conclusion of the MEC-SMAP is highlighted for its SP-based session establishment while the formation of the SP along with its standardization on the layers is presented. In addition, the security application model, security cost computation model, and the impact of the SP switching cost are further elaborated. Section 6.3 describes the formulated Markov Chain-based estimation model for determining the predicted SP to be applied, while the model statistics and its validation results are mentioned afterward. The developed prototype MEC-SMSF is described in Section 6.4 along with its specifications, while the details of the conducted emulations are specified. Further, the timing KPIs of the MEC-SMSF are benchmarked while the migration delay is measured for different conditions. Section 6.5 summarizes the chapter content while presenting the concluding remarks.

6.1 Introduction

6.1.1 Motivation

In accordance with the Use Case specified in Section 1.1.2, the latency associated with the link between the MEC edge and the UE is imperative for service continuity [184]. In a handover situation where the UE is shifting from one serving gNodeB (gNB) to another, the communication channel latency would not be affected due to the existence of soft-handover scenarios. Though, the corresponding service should be migrated to the roamed gNB for maintaining service continuity. A typical service migration takes around 1 to 5 s or more for service initiation [21]. If the service migration process is initiated at the handover commencing instant, the migration and configuration (i.e. installing and configuring the service instance in the novel gNB environment) latency is exceeding the time taken for the handover process. Thus, the delay associated with the migration process is a factor that should be contemplated before realizing low-latency-prone applications. Therefore, alleviating the latency is a prime requirement for all priority MEC-enabled services, especially AV-based deployments.

Security is another important aspect to be considered with service migration phenomena. As migrations are transferring executable content, and thereafter would be configured and integrated to the virtualization platform, a successful instilling of malicious content could expose the entire MEC edge environment via resource exhaustion strategies [89]. Though, security in current times is not an assurance that can be guaranteed by a single security mechanism. At the least, basic confidentiality, integrity, and availability aspects should be covered by any action committed by novel systems. Thus, several security mechanisms should be amalgamated to assure the security of a communication channel. In the case of service migration, with its requirement for avoiding penetration threats, a formidable security level should be applied to the migration protocol. This is causing a conflict between the unprecedented security cost (i.e. time for security processing and the overhead), and managing the latency in accordance with the Service Level Guarantees

(SLGs) of emerging services. The exploitation of this relationship is a vital conundrum for service migrations, especially for AV applications. Thus, this chapter presents a model for optimizing the security application and latency mitigation of service migration channels.

6.1.2 Related Literature

Service migration of edge computing is not a relatively novel topic. In fact, research conducted on service migration is mostly targeted at migrating models formed to optimize energy consumption using either Markov or Lyapunov optimization techniques [1]. In such various areas, service migrations in vehicular applications are quite available, as in [185, 186, 187, 188, 189, 190]. A mobility-aware Blockchain-enabled multi-side offloading scheme for vehicular fog cloud networking was introduced by Lakhan et al. in [191], reaffirming the requirement for mobility-aware security in a VANET context. Further, Xiao et al. in [192] propose a coalition-game-based and location-aware approach to MEC service migrations in mobile user reallocation in crowded circumstances to manage optimal resource utilization. Despite their valuable contributions towards modeling the migration problem with the dynamic nature of vehicular agents, security is not a prime consideration.

Zhang et al. in [151] propose Falcon, a blockchain-based service migration framework for edge computing. The migration process has been made flexible with the use of VMs or containers as mobile agent-based carriers. A selection algorithm has been designed to maximize the migration benefits in line with the service quality. The immutable alliance chain decentralized to edge clouds is improving the performance of the Falcon with blockchain inclusion. The blockchain-based deployments however aren't the solutions for most use cases or applications. Cui et al. in [157] introduce a jamming strategy utilizing fountain codes for service migration scenarios. A set of Relay nodes should be maintained to conduct cooperative jamming, which eventually misleads the eavesdropper and deteriorate the sensing quality of the migration channel. This strategy, however, does not provide a solution for trust/ authenticity verification among migration entities. In fact, these approaches do not address

the drawbacks that security applications can cause in the context of priority services.

Wang et al. in [193] presents a service migration model for MEC, with accurate sensing of user location privacy, which is ideal for vehicular applications due to its reliance on location awareness. The authors model the total cost of the system aggregating migration cost, user-perceived delay, and the leakage of location privacy. As in most cases, the service migration problem was modeled as a Markov Decision Process (MDP). The conducted simulations support the proposed MDP strategy in minimizing the long time overall cost of the system, which includes the privacy leakage metrics. Though this approach reduces the eavesdroppers tracking ability of UE location, the directive does not bare any relation to the optimization of security level for improving service efficiency. To the best of my knowledge, current literature lacks such a research directive.

6.1.3 Contribution

Therefore, the main contributions of this chapter can be listed as:

- Introducing and validating the concept of dynamic security via Security Profiles for applying consolidated security mechanisms, which enables the standardization of security for service migration channels and paves the path for security management with SLG objectives.
- Formulating a latency-associated cost model for security mechanisms, that benchmarks the security mechanisms based on their encryption, decryption, and overhead costs.
- Proposing and validating a model for estimating the most probable security profile/setting utilizing Makov chains on the service migration channel dynamics, for reducing the switching delay between security profiles.

The Tables 6.1 and 6.2 tabulate the notions and acronyms used in this chapter.

Table 6.1: Main Notions and Acronyms with their Definition I

Acronym	Definition
MEC Specific	
gNB_X	gNodeB of the X party
UE	User Equipment
g2g	gNB-to-gNB
SMC	Service Migration Channel
MES	Mobile Edge Service
gNB_S	Source gNB
gNB_R	Roaming gNB
BW	Bandwidth
Security Specific	
K_M	Migration Key
SyK_{XY}	Symmetric Key between X and Y
K'_M	Normalized migration key
m, m'	fragmented migrating content and processed (encrypted) content
$CIPHER$	Indicates the encryption-based configuration metric
$HASH$	Hash-based configuration metric
SEC_x	Specify the x security mechanism
$ENC()$	Encryption function
$DEC()$	Decryption function
IV	Initialization vector of a security algorithm
Security Profile and Cost Model Specific	
SP	Security Profile
spi_s	Security profile index of a SP
l_x	layer x of the SP
$l_x()$	Function for applying the security mechanism specific to l_x
SS	Security Setting: Sequence of applying SPs
$\mathfrak{M}$	Migration content
$\mathfrak{N}$	Normalization function
ω	Bandwidth
ω_R	Residual or unused BW of a channel
T_M	Migration time
T_E	Encryption time
T_D	Decryption Time
θ	Security overhead/ digest

Table 6.2: Main Notions and Acronyms with their Definition II

Acronym	Definition
Security Profile and Cost Model Specific	
$fragment()$	Function for fragmenting content of larger sizes
$threshold()$	Function for identifying the thresholds for SP switching
$extract()$	Function for sensing the BW of the channel
$AGG()$	Aggregating processed messages to form m'_i
$SWITCH()$	Function for switching the SPs
$SEND()$	Sending a message via the SMC
$ALGO$	Algorithm
I	Input size of the plaintext
D	Divisor
$MAP()$	Function that map the optimal SP for current ω_R
Markov Model Specific	
PS	Priority service
NS	Non-priority service
M	Number of channels in the network
B	Bandwidth of a channel
W, V	Lower and upper bound of the number of aggregated channels by a non-priority service
$\mathcal{S}$	The set of feasible states of the system
Q	Transition rate matrix of the CTMC
π_i	Steady-state probability of being in state i
$\boldsymbol{\pi}$	Steady-state probability vector

6.2 Dynamic Security Management

The requirement for dynamic security is established from the explication presented in Section 6.1.1. If the security credentials of a security protocol enabled for a specific service migration session remain the same for the entirety of the process, there is a possibility of overwhelming the bandwidth utilization of the Service Migration Channel (SMC), due to the higher cryptographic overhead/ digest generated from the higher level cryptographic algorithms. In such a circumstance, the Transceiver (TRX) of the SMC would be forced to limit its data rate. Hence the service migration process is delayed and could cause a service disruption in case the traversing UE reaches the next MEC-enabled gNB domain prior to the conclusion of the migration process. The conse-

quences of such disruption are severe for VANET-based use cases specified in Section 1.1.2. Despite the fact that the possibility of occurring such a situation where the SMC is already highly occupied with other traffic is lesser, envisaged traffic models with denser traffic in metropolitan areas for emerging mobile technologies [194, 195] and the severity of the consequences of AV deployments [196] are encouraging this concept of dynamic security that enables maintaining service continuity. This concept reaches beyond just changing the security credentials or cryptographic keys of a certain security protocol. In fact, the utilized security mechanisms or algorithms can be changed during the migration interval. The intention of this research is to introduce the concept of Security Profile (SP), which can standardize dynamic security management as a tool for overall security management in future mobile networks.

6.2.1 Recalling MEC SMAP Mutual-Authentication and Security Profile Establishment

In the authentication phase presented in Chapter 5, the main functions are 1) validation of the gNB identities, 2) determining the legitimacy/ integrity of the migrating MES, and 3) launching capability of the migrating MES at the roaming gNB MEC environment. The authentication phase is concluded by securely transferring the credentials and parameters corresponding to computing the migration master key K_M. This is a Perfect Forward Secrecy (PFS) enabled tamper-proof key generated for each migration session, unique to the gNBs participating in the migration.

Fig. 6.1 illustrates the high-level perspective of the mutual authentication scheme and the security profile selection process. The security management function is carried out through the utilization of Security Profiles (SPs) explained below. After the initial gNodeB-to-gNodeB (g2g) mutual authentication scheme, a set/ array of SPs will be conveyed to the roaming gNB in a message secured with SyK_M, which is a symmetric key. Depending on gNB_R's instantaneous processing power and the affordable BW utilization for the migration channel, an appropriate SP will be selected. This selection is supported by the security cost parameter of each SP computed for the channel BW, ex-

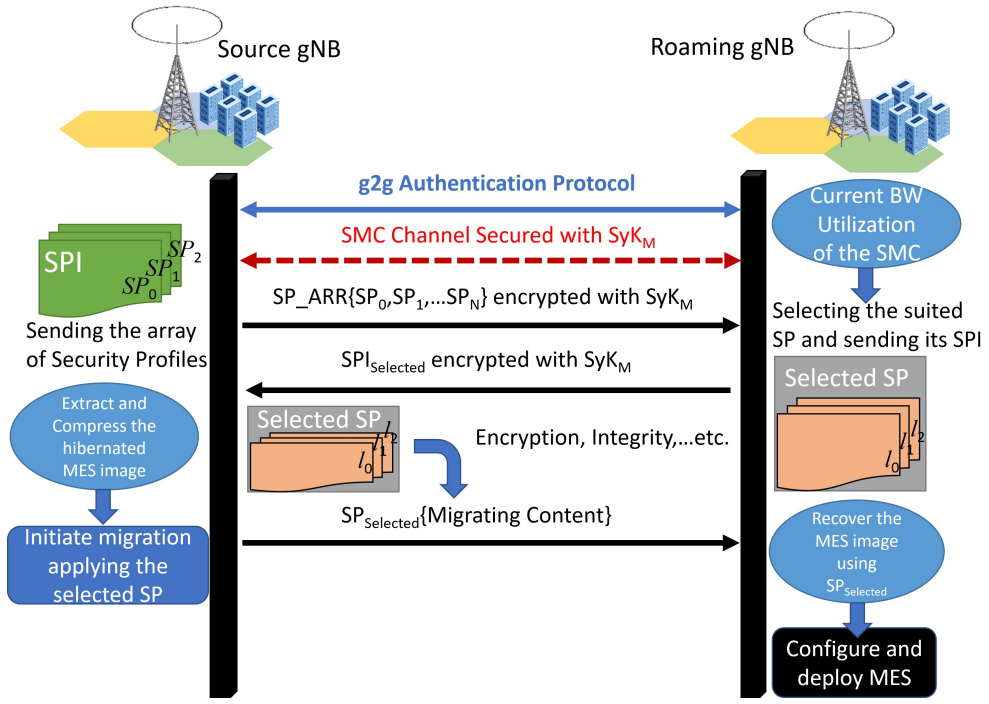


Figure 6.1: High-Level Mutual Authentication Process and its Conclusion

plained later in this chapter. The security profile index of the corresponding SP_S (i.e. selected SP) will be sent to the gNB_S; upon receiving, gNB_S will configure the SP with the migration key K_M. Then the service migration process is initiated by extracting a hibernated version of the MES container image. This image is then compressed and sent through the channel secured with the applied $SP_S(K_M)$. At the receiving end, gNB_R will recover the encrypted migrated content from the same $SP_S(K_M)$. The recovered content will then be de-compressed and configured to be deployed in the gNB_Rs MEC environment.

6.2.2 Security Profile (SP)

As stated above, a full assurance of security cannot be attained via a singular security mechanism (i.e. encryption alone). In hybrid security protocols, more than one security mechanism is involved in securing a channel. Thus, a standard template should exist to classify and specify the existing security mecha-

nisms; and the defined SP is the template for achieving this standardization. A global standardization would allow security compliance over the global terrain for new technologies that eliminate the obvious interoperability, and compatibility issues governing multi-vendor environments. A SP is represented by different security layers that accompany signaling, encryption, integrity, PFS, timing, and any other additional security measures. As indicated in Table 6.3, such aspects can be specified and defined under the concept of the SP, where layering offers a way to classify and group security mechanisms, so as to prioritize them in accordance with the circumstance.

Table 6.3: Specifications of the Proposed Security Profile Standard

L.No.	Layer Index	Layer Name	Components	Example
0	SI	Signalling Channel	ALGO, K_M, SyK_M, key size	AES, 256b
1	EN	Encryption	ALGO, key size, PADDING scheme, block cipher mode	AES, RC4, 128b, 256b
2	IN	Integrity	ALGO, version, digest size, block cipher mode	SHA, MD, Keccak, 128b, 256b, 512b
3	TI	Timing	TS format, security association lifetime, approved clock skew	ms, ns, 3600 s, 1 ms
4	FS	Forward Secrecy Standing	FS standing, Key dissemination method/ standard, key size	PFS, ECDH, RFC5903
5	AD	Additional security measures	DoS/ DDoS Repellent, QR resistance mechanisms	DoS puzzle, QR lattice

Each SP is specified with a unique identifier, Security Profile Index (SPI), that makes the operations of selection and application more convenient. Even though the SP concept defines a standardization that can be extended globally, the management of the SPs should be conducted locally, as the exposure of a specific SP intentionally or otherwise could compromise the entire

security/ trust domain. Thus, SP creation, dissemination, and management can be maintained peer-to-peer, or within a specific domain. Each layer of the SP as specified in Table 6.3, contains parameters or components. Such components specify the algorithm, embedded key(s) size, padding scheme and size, block cipher mode, integrity checking mechanism, tunneling scheme, and other additional security schemes (i.e. for DoS mitigation, jamming as in [157]). Each $SP(K_M)_i$ corresponds to a different set of layers $l_0, l_1, l_2, ...l_j$, where j could represent values from 0 to any number. The depth of layers (i.e. j) and its applied mechanism will determine the security level (L) and the incurred costs in terms of computation ($COMP$) and communication ($COMM$) perspectives. $COMM$ cost is the resulting BW utilization each $SP(K_M) - l_i$ is deploying, while $COMP$ accounts for the average processor utilization for each $SP(K_M) - l_i$ processing.

6.2.3 Security Management

Selecting an optimal SP depending on the BW utilization at the migration initiation alone won't ensure the maximal service level efficiency for this channel. As a typical servicing container images (i.e. Busybox or OpenFace) can take migration times excessive of 7 s with considerable downtime; dynamic and unprecedented events associated with the UEs interfacing the gNB, or other MES migrations can trigger changes to the instantaneous BW utilization within the migration period [152]. If a significant reduction in the current remaining channel BW is to ensue, the SP_S would still exhibit an alleviated performance with an aggregated latency to the migration time. This latency might be critical in the context of 5G or B5G applications such as AVs. Thus, to mitigate such occurrences, continuous real-time monitoring of the migrating channel should be conducted to record the residual BW. With this parameter, SPs can be switched accordingly to either reduce the security level while alleviating the otherwise applicable latency or improve the security level in case the residual BW increases. This approach is enabling the maximization of the applicable security level for a given residual BW of the SMC. This leads to proper managing/ optimizing of the security in the communication channel. The method of

SP security application and the formulation of the security cost model follows in this sub-section.

6.2.3.1 Security Application Model of the SP

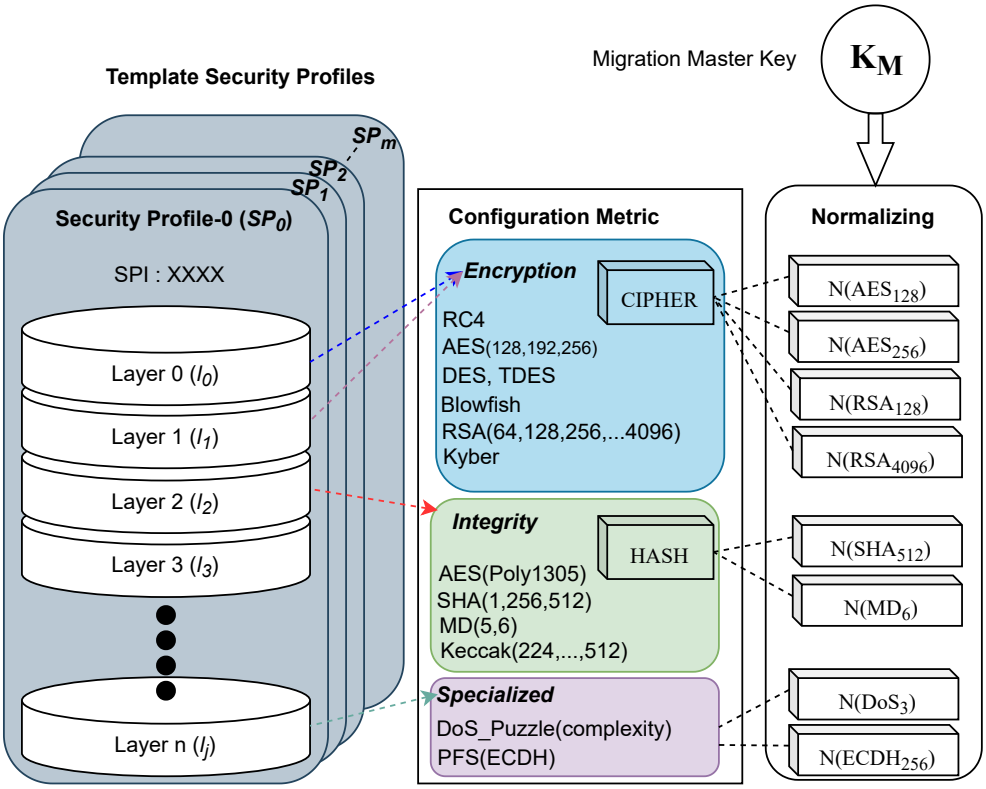


Figure 6.2: Model of the Security Profile and its Context

As indicated in Fig. 6.2, the configuration of a SP is dependent on each security mechanism or algorithm. In the figure, the configuration metric classifies the functions of the available security mechanisms, as in encryption, integrity, or any other specialized direction indicated in Table 6.3. In fact, the configuration metric specifies which primitive of the engaged security mechanism (i.e. $CIPHER, HASH, SEC_0, ..., SEC_y$) should be updated with the K_M. Though, K_M itself cannot be substituted as the keys, Initialization Vectors (IVs), secrets, or any other key parameter that enables the security mechanisms in the SP. Thus, K_M should be amended or transformed based on

the nature and size of the cryptographic primitive. This transformation process is Normalization ($\mathcal{N}$). Each normalization function is specific to the subjected algorithm or mechanism. Typically for encryption algorithms, K_M can be configured as the shared secret. In that case, $\mathcal{N}(CIPHER)$ represents cropping or duplication of K_M till it resembles the shared secret length. For integrity, IV or the Salt, or both can be replaced by $\mathcal{N}(HASH)$ process. Similarly, other mechanisms can be defined with their own $\mathcal{N}$ function.

Algorithm 1: Method of applying the security mechanisms specified in SP for the message $\mathcal{M}$

Input: $spi_s, K_M, \omega_R, \mathcal{M}$
Output: $T_M(\mathcal{M})$
Init: $fragment(\mathcal{M}) \equiv [m_0, m_1, m_2, ..., m_i, ..., m_x]$
$threshold(\omega_R) = [\omega_t^{min}, \omega_t^{max}]$
Record T_{init} **while** $i \leq x$ **do**
  $\omega_R^{new} \leftarrow extract(\omega_R)$ **if** $\omega_R^{new} < \omega_t^{min}$ *OR* $\omega_R^{new} > \omega_t^{max}$ **then**
    $spi_c^{new} \leftarrow MAP(\omega_R^{new})$ Update $threshold(\omega_R^{new})$
  end
  if $spi_c = spi_s$ **then**
    $\mathcal{N}(spi_s, K_M) \equiv [K'_{M,0}, K'_{M,1}, ..., K'_{M,j1}]$
    $SP(spi_c, K'_{M,0}) \rightarrow l_0(m_i) \equiv m'_{i,0}$
    $SP(spi_c, K'_{M,1}) \rightarrow l_1(m_i) \equiv m'_{i,1}$
    $SP(spi_c, K'_{M,2}) \rightarrow l_2(m_i) \equiv m'_{i,2}$
    ..... $SP(spi_c, K'_{M,j1}) \rightarrow l_{j1}(m_i) \equiv m'_{i,j1}$
    $AGG(m'_{i,0}, m'_{i,1}, m'_{i,2}, ..., m'_{i,j1}) \rightarrow m'_i$ $SEND(m'_i)$
  else
    $SWITCH(spi_s^{new})$
    Update $spi_c \leftarrow spi_c = spi_s^{new}$
    $\mathcal{N}(spi_s^{new}, K_M) \equiv [K''_{M,0}, K''_{M,1}, ..., K''_{M,j2}]$
    $SP(spi_c, K''_{M,0}) \rightarrow l_0(m_i) \equiv m'_{i,0}$
    $SP(spi_c, K''_{M,1}) \rightarrow l_1(m_i) \equiv m'_{i,1}$
    $SP(spi_c, K''_{M,2}) \rightarrow l_2(m_i) \equiv m'_{i,2}$
    ..... $SP(spi_c, K''_{M,j2}) \rightarrow l_{j2}(m_i) \equiv m'_{i,j2}$
    $AGG(m'_{i,0}, m'_{i,1}, m'_{i,2}, ..., m'_{i,j2}) \rightarrow m'_i$ $SEND(m'_i)$
  end
end
Record T_{end}
Compute $T_M(\mathcal{M}) = T_{init} - T_{end}$

The Algorithm 1 describes the security management process for the compressed migrating content $\mathcal{M}$, which is fragmented to the array of $[m_0, m_1, m_2, ..., m_i, ..., m_x]$. The ω_R represents the residual BW of the migrating channel. At the initiation of the algorithm, apart from the fragmentation, thresholds of ω_R are established. When the updated ω_R reaches beyond either the min or max thresholds, the current SP should be switched. Further, the time of the migration initiation is recorded as T_{init}. For each repetition of the algorithm, the K_M is normalized for each layer of the selected SP_S. The m_i is being subjected/ processed by the security mechanisms represented by $l_j(m_i)$ function that outputs $m'_{i,j}$. After all the security mechanisms are applied, each outcome is aggregated (i.e. AGG) to form the secure message content m'_i, which will then be sent to the gNB_R. The algorithm describes the security application process at both instances of SP being switched and retained. In the end, migration time T_M is computed for this migration session. Algorithm 1 only explains the gNB_S operation, and gNB_R operates inversely to recover the compressed migrated content. This is algorithm is implemented in the developed prototype MEC-SMSF deployment for its security application process available under Appendix B.5.

6.2.3.2 Formulated Security Cost Model

$$C_{security}(SP, K_M) = \sum_{x=1}^{j}[(T_E + T_D)_{l_x}] + \frac{1}{\omega_R}\sum_{y=1}^{j}\Theta_{l_y} \tag{6.1}$$

The formalization of the proposed SP selection process is entirely reliant on its quantification approach, and the existence of a bench-marking method. In order to benchmark, the cost of each SP should be determined; more precisely, cost of the each respective layer of the SP. The Eqn. 6.1 specifies the formulated cost function for a specific security profile, SP. The security cost ($C_{security}$) is a latency-associated representation that is amalgamating the values of encryption time (T_E), decryption time (T_D), and the security overhead (Θ) induced latency corresponding to each layer of the SP. As this research directive is an attempt to exploit the relationship between security and its associated latency, the $C_{security}$ was designed to formulate a latency dimension.

Algorithm 2: Algorithm for Computing the Security Cost of a Security Profile

Input: $spi, K_M, \omega_R, \mathcal{M}$
Output: $C_{security}(spi, K_M)$
while $i \leq j$ **do**
  $l_i \rightarrow CIPHER = (0,1), HASH = (0,1), SEC_0 = (0,1), ..., SEC_y = (0,1)$
  if $CIPHER = 1$ **then**
    Recall $ALGO \leftarrow SP(spi, l_i)$
    $t_E \leftarrow ENC(ALGO(K_M, \mathcal{M}))$
    $\theta \leftarrow ENC(ALGO(K_M, \mathcal{M}))$
    $t_D \leftarrow DEC(ALGO(K_M, \mathcal{M}))$
  end
  if $HASH = 1$ **then**
    Recall $ALGO \leftarrow SP(spi, l_i)$
    $t_E \leftarrow ENC(ALGO(K_M, \mathcal{M}))$
    Digest, $\theta \leftarrow ENC(ALGO(K_M, \mathcal{M}))$
  end
  if $SEC_0 = 1$ **then**
    Recall $ALGO \leftarrow SP(spi, l_i)$
    $t_E \leftarrow ENC(ALGO(K_M, \mathcal{M}))$
    $\theta \leftarrow ENC(ALGO(K_M, \mathcal{M}))$
    $t_D \leftarrow DEC(ALGO(K_M, \mathcal{M}))$
  end
  .....
  if $SEC_y = 1$ **then**
    Recall $ALGO \leftarrow SP(spi, l_i)$
    $t_E \leftarrow ENC(ALGO(K_M, \mathcal{M}))$
    $\theta \leftarrow ENC(ALGO(K_M, \mathcal{M}))$
    $t_D \leftarrow DEC(ALGO(K_M, \mathcal{M}))$
  end
  $C(l_i) = t_E + t_D + \frac{\theta}{\omega_R}$
end
$C_{security}(spi, K_M) = \sum_{x=1}^{n}[C(l_x)]$

Even though there are other factors such as processing power and transmission energy associated with a SP that can be included in this cost model, in this setup involving the two gNBs and a shared migration channel, such factors are common for all the contrasting SPs, and therefore can be neglected from this formulation. Further, a MEC environment that possesses the required computational resources is assumed and is only constricted by the communication resources indicating a heavy tele-traffic domain.

The Algorithm 2 specifies the model for computing the $C_{security}(spi, K_M)$ for a particular SP. As specified in Fig. 6.2, each layer of the SP has distinct security metrics, that could either be a $CIPHER$ (i.e. for encryption), $HASH$ (i.e. for integrity verification), or any other SEC_y mechanism. Each $CIPHER$ function, depending on their $ALGO$ (i.e. security algorithm) produces different or null t_E, t_D, and θ values. However, a layer can possess a single security metric only. And the term $l_i \rightarrow CIPHER = (0, 1)$ indicates whether the specified security metric is applied for the given l_i layer or not. Other metrics operate in the same way. Thus, Algorithm 2 computes the t_E, t_D, and θ for each layer, resulting in $C(l_i)$, until the depth j is reached, and finally determines the $C_{security}(spi, K_M)$ through aggregation.

As stated earlier t_E, t_D, and θ values are security mechanism-specific values, and should be pre-computed and available for different specifications of the security mechanism. The research conducted in [197, 198] provides some context on the cost for certain algorithms such as AES-GCM-256 employed in BoringSSL, Libsodium, and Crypto PP. For some $ALGO$s a function can be derived for the variation of t_E, t_D, and θ values. As an instance, Fig. 6.3 indicates the functions for computing overhead digest (i.e. θ) of AES algorithm for its Cipher Block Chaining (CBC), Electronic Code Book (ECB), and Counter (CTR) block cipher modes along with its padding schemes. However, these functions were derived from the observations gathered from the conducted experiments. The formation of this experimental setup is discussed under Section 6.4.1 of this chapter. Even though the cryptographic overhead or digest size is independent of the processing hardware, encryption and decryption times are obviously dependent on the resources of the MEC environment. Further, the variation of the AES overhead with the input or plaintext size

CBC/ECB with PKCS5Padding

$\theta_{min} = 24$

$\Delta = 20$

$D = 48$

$\theta_{max} = \theta_{min} + 2.\Delta$

$$\hat{I} = \begin{cases} 0, 0 < I < 48 \\ \left\lfloor \frac{I}{D} \right\rfloor, I \geq 48 \end{cases}$$

$$\bar{I} = \begin{cases} I, 0 < I < 48 \\ I_{rem} : [D = x.I + I_{rem}, x \in \mathbb{N}], I \geq 48 \end{cases}$$

$\hat{\theta} = \theta_{max} \times \hat{I}$

$$\theta = \begin{cases} \hat{\theta} + \theta_{min}, 0 < \bar{I} < 16 \\ \hat{\theta} + \theta_{min} + \Delta, \bar{I} < 32 \\ \hat{\theta} + \theta_{min} + 2.\Delta, \bar{I} \geq 32 \end{cases}$$

CBC with NoPadding

$\theta_{min} = 24$

$\Delta = 20$

$D = 48$

$\theta_{max} = \theta_{min} + 2.\Delta$

$$\hat{I} = \begin{cases} 0, 0 < I < 64 \\ \left\lfloor \frac{I}{D} \right\rfloor, I \geq 64 \end{cases}$$

$$\bar{I} = \begin{cases} I, 0 < I < 64 \\ I_{rem} : [D = x.I + I_{rem}, x \in \mathbb{N}], I \geq 64 \end{cases}$$

$\hat{\theta} = \theta_{max} \times \hat{I}$

$$\theta = \begin{cases} \hat{\theta} + \theta_{min}, (\bar{I} = 16 \wedge \lfloor I/D \rfloor = 0) \\ \hat{\theta} + \theta_{min} + \Delta, (\bar{I} = 32 \wedge \lfloor I/D \rfloor = 0) \\ \hat{\theta} + \theta_{min} + 2.\Delta, (\bar{I} \geq 32 \wedge \lfloor I/D \rfloor = 0) \end{cases}$$

CTR with any Padding

$\theta_{min} = 4$ $\quad D = 3$ $\quad \hat{I} = \left\lfloor \frac{I}{D} \right\rfloor$ $\quad \theta = \hat{I} \times \theta_{min}$

I - Input Size

θ - Output/ Digest Size

Figure 6.3: Model for the AES Cryptographic Digest

is plotted in Fig. 6.4 for different block cipher modes and padding schemes.

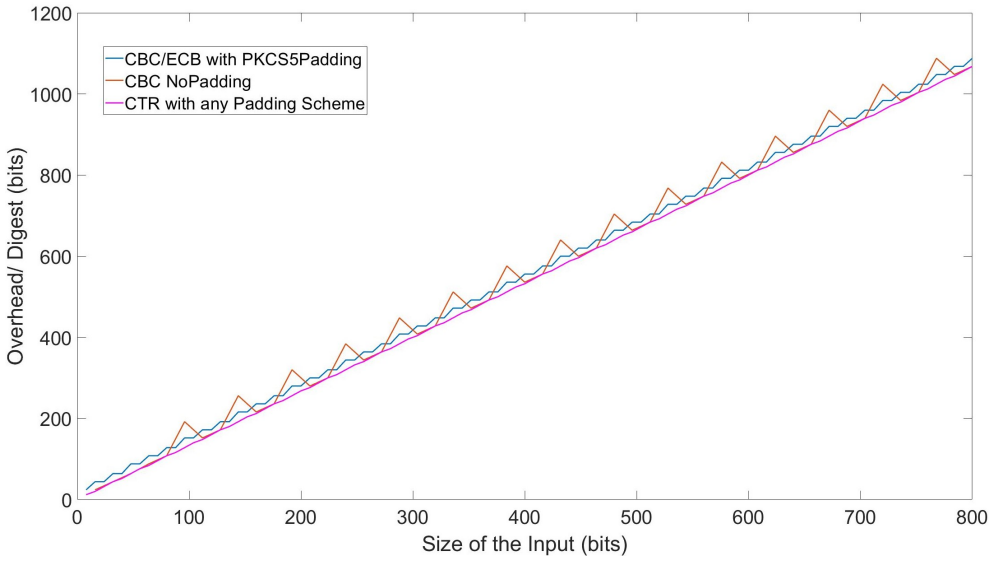


Figure 6.4: AES Overhead Cost Variation

Conversely, an experiment was conducted to record the variation of t_E, t_D, and θ parameters for the algorithms of AES-128, AES-192, AES-256, RC4,

and BF; which are the most widely used security algorithms for data transmissions in modern protocols. According to Fig. 6.5 and Fig. 6.6, t_E and t_D depict an approximated linear variation for all the schemes, and AES is efficient and less time-consuming for higher payload sizes. Fig. 6.7 depicts that RC4 and BF ciphers generate a lesser cryptographic digest contrast to AES.

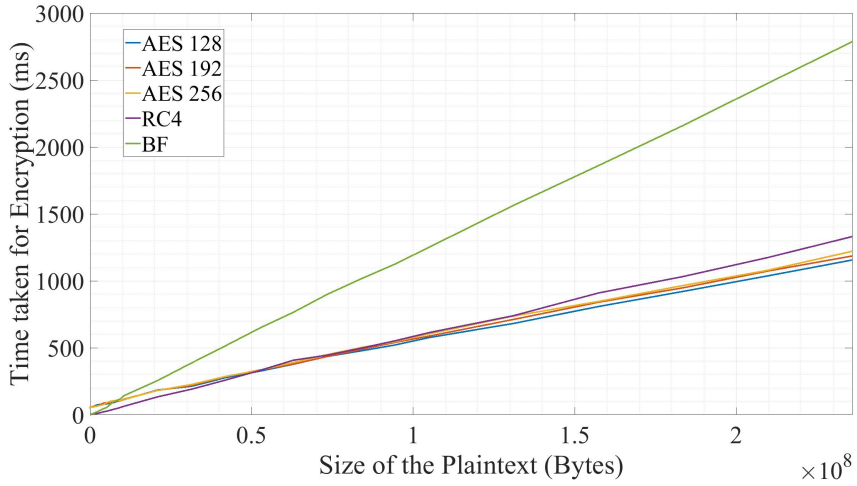


Figure 6.5: t_E Variations of AES, RC4, and BlowFish Security Algorithms

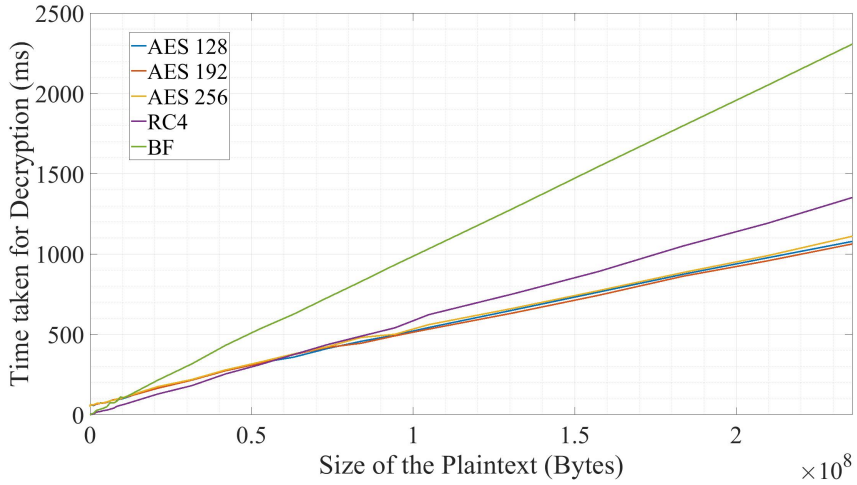


Figure 6.6: t_D Variations of AES, RC4, and BlowFish Security Algorithms

In a similar manner, experiments were conducted for Keccak, SHA, MD5,

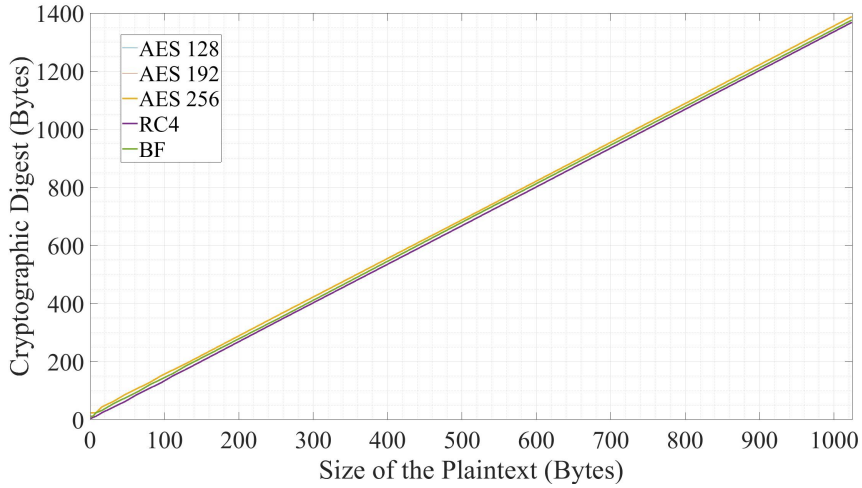


Figure 6.7: θ Variations of AES, RC4, and BlowFish Security Algorithms

and Kyber (i.e. for quantum-resisting cryptography) algorithms under their various key sizes and other settings. It is perceived that it is possible to derive cost functions as in AES for most other algorithms. The security mechanisms/ algorithms widely used for data transmissions were considered for the formed experiments as the scope is limited to the service migration aspect.

6.2.3.3 Impact of Security Settings and the SP Switching Cost (SC) for Service Migration

Maximum security settings or application of the highest security profile is possible when the migration channel has an affordable ω_R. With critical stages of the BW utilization (i.e. above 80% utilization) though, the impact caused by each security algorithm/mechanism is vital. As seen from Fig. 6.8, migration time variation for BW utilization above 70% in the SMC has been plotted for different security settings, of migrating content that is 16GB in size. As expected, the plot suggests an exponential increment in the T_M for BW utilization above 95%. An attempt to switch the security algorithms was simulated in scheme 1 and scheme 2, where security was shifted from the AES block cipher to the RC4 stream cipher. Though the simulation suggests the smooth operation of the SP switching, a significant switching cost (t_{SC}) is an obvious

expectation. t_{SC} accounts for the additional time taken from SMC sensing, detection of exceeding the $threshold$, to performing the $SWITCH$ operation as seen in Algorithm 1. Thus, minimizing the t_{SC} is an imperative aspect of the proposed dynamic security model.

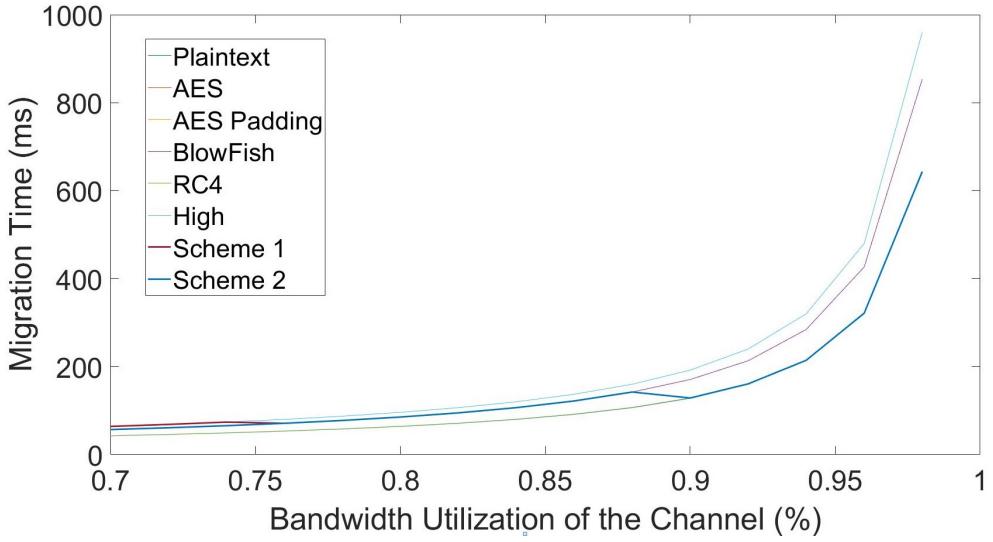


Figure 6.8: Variation of the Migration Time for Different Security Settings/ Algorithms

6.3 Proposed Dynamic Security Estimation Model

In ideal circumstances where the real-time sensing of the SMC is plausible, the proposed model performs well as seen in Fig. 6.8. Though, constraints on the availability of resources for channel sensing and performing the $SWITCH$ operation limit the validity of the model, as t_{SC} could reach beyond the limits of the gained advantage from demoting the security settings. Prior determination of the SPs to be used during the migration process, however, will improve this situation. Prediction and estimation are not novel adoptions for service migration concept as exhibited from the prevailing literature [199, 193, 145]. Markov models are widely used for such estimation tasks specifically where the channel states are involved and are adopted for both continuous and

stochastic environments. Thus, such Markovian approaches are employed for prior-estimating the required SP, hence minimizing the t_{SC}. The proposed methodology is outlined in Fig. 6.9, where the intention is to estimate the most probable security mechanism/setting for the estimated bandwidth utilization of the SMC.

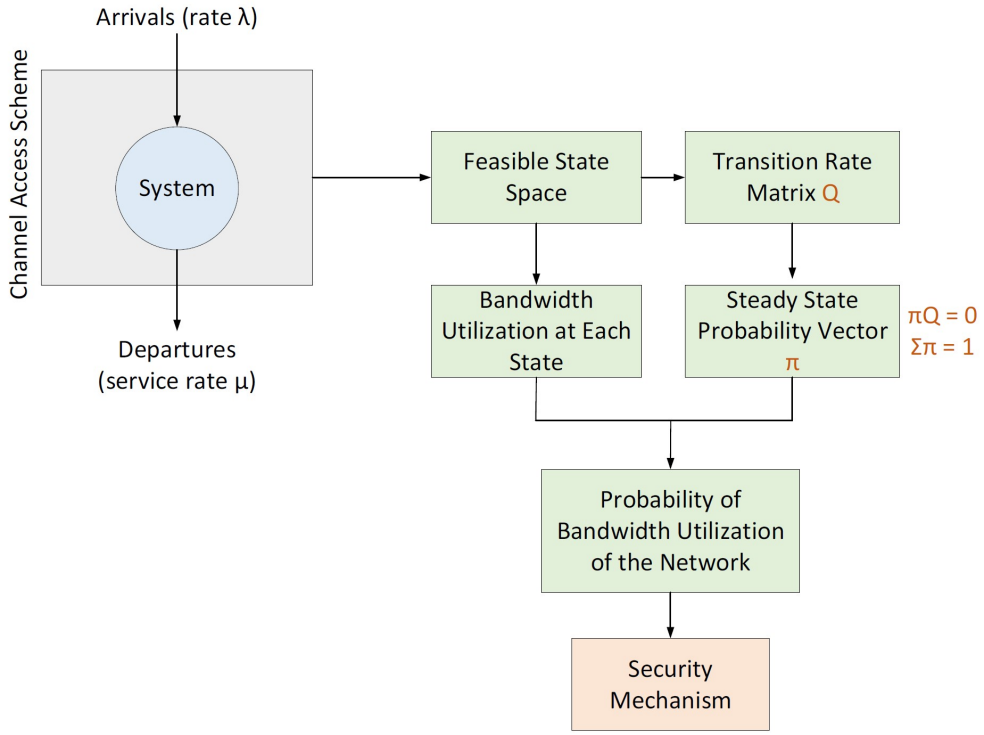


Figure 6.9: Proposed Security Estimation Model

In this section, the proposed dynamic security estimation model is presented consisting of the dynamic channel access scheme, the corresponding continuous-time Markov chain (CTMC) model, and the security estimation scheme. Contrary to the popular approaches of employing Markovian approaches for service migration decision-making, the following strategy in this research involves Markov chains for determining the optimal SP for conducting the migration.

6.3.1 Dynamic Channel Allocation Scheme and the Continuous Time Markov Chain Model

Studying the Dynamic Channel Allocation (DCA) strategy proposed in [200], a CTMC is formulated to generate a feasible state space of the considered network scenario. In this DCA scheme, two types of services are considered for specifying MESs, namely, priority services (PS) and non-priority services (NS). The arrivals of both PS and NS services are Poisson processes with arrival rates λ_P, λ_S respectively. The service times for PS and NS services are exponentially distributed. The service rates per channel for PS and NS services are denoted μ_P and μ_S respectively.

Consider a system with M non-overlapping channels with equal bandwidth B. In the following, the channel access procedure adopted in [200] is revisited with respect to the PS and NS activities.

6.3.1.1 Access Scheme

When a new NS request arrives, the system should offer at least W channels to accommodate the new arrival. Otherwise, the request is rejected. Moreover, the new NS may aggregate up to V, $(V \geq W)$ channels if available. If the number of idle channels upon an NS arrival is fewer than W, the ongoing NS with the maximum number of channels will donate one or several channels as long as it can still have W channels after donation. If the number of idle channels plus the number of channels that can be donated by ongoing NS services is still fewer than W, the new NS request is blocked.

If a new PS arrives at a moment when there are idle channels in the CRN, the new PS can start transmission in an idle channel. If there is no idle channel, the NS which has the maximum number of aggregated channels will donate one channel to the interrupted NS, given that it has more aggregated channels than the interrupted NS and its remaining number is still not fewer than W. In the worst case, if all ongoing NSs have exactly W channels upon a PS arrival, the interrupted NS service is forced to terminate. On the other hand, upon a service completion, the ongoing NS which has the minimum

number of aggregated channels will utilize those idle channels up to V.

Table 6.4: Selected Critical Bandwidths

State	Description
BW1	BW utilization less than 87.5%
BW2	BW utilization = 87.5%
BW3	BW utilization = 91%
BW4	BW utilization = 93.7%
BW5	BW utilization = 96.8%
BW6	BW utilization = 100%

6.3.1.2 Steady State Probability Calculation

The states of the CTMC model corresponding to the DCA scheme can be represented by $\boldsymbol{x} = (i_{ps}, j_W, j_{W+1}, \cdots, j_V)$ where i_{ps} denotes the number of PS services in the system. The number of NS services with k aggregated channels is denoted as j_k where $k = W, W+1, \cdots, V$. Let $\mathcal{S}$ be the set of feasible states of the system such that $\mathcal{S} = \{\boldsymbol{x} | i_{ps}, j_W, j_{W+1}, \cdots, j_V \geq 0; b(\boldsymbol{x}) \leq M\}$ where $b(\boldsymbol{x}) = i_{ps} + \sum_{k=W}^{V} k j_k$.

Let the steady state probability of being in state i is denoted as π_i. It can be calculated from the global balance equations and the normalization equation, which are given as

$$\boldsymbol{\pi Q} = \mathbf{0}, \ \sum_{i \in \mathcal{S}} \pi_i = 1, \tag{6.2}$$

where $\boldsymbol{\pi}$ is the steady state probability vector, $\boldsymbol{Q}$ denotes the transition rate matrix, and $\mathbf{0}$ denotes a row vector of 0's.

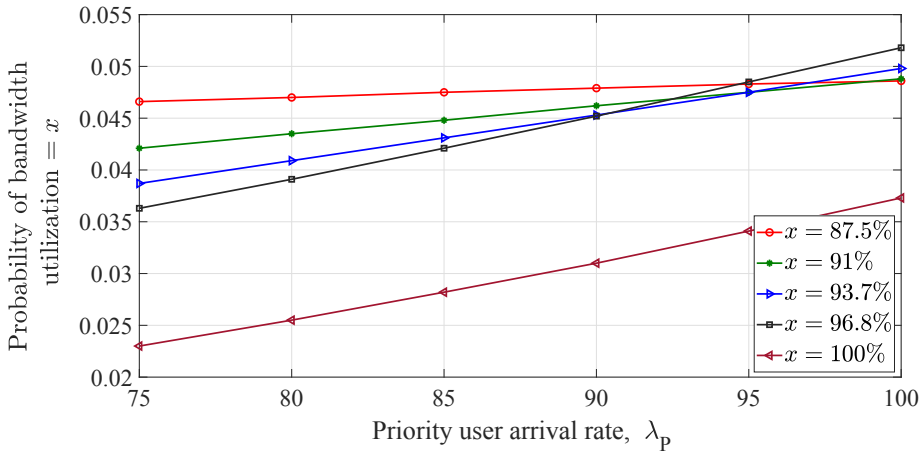


Figure 6.10: Relationship of Bandwidth Utilization and Arrival Rate of the created CTMC model

The created steady-state CTMC model with the default parameters of M = 32, $\mu_P = 5$, $\mu_S = 10$, $\lambda_P = 75$, and $\lambda_S = 150$, generates the probabilities of critical bandwidths depicted in Fig. 6.10 for different priority arrival rates. The selected critical bandwidths are tabulated in Table 6.4. Fig. 6.10 shows that the probability of having x bandwidth utilization rises when the user arrival rate λ increases. However, the intensity of the increase in bandwidth utilization probability depends on the x value. Therein, a low bandwidth utilization level (for instance $x = 87.5\%$) does not exhibit a rapid increase with λ values compared to a high bandwidth utilization level (for instance $x = 96.8\%$).

6.3.1.3 Embedded Markov chain

With a continuous-time Markov process X_t, a discrete-time Markov chain (DTMC) can be associated, with the so-called embedded Markov chain. Let $p_{s,s'}$ and $q_{s,s'}$ denote the probability that the system makes a transition from state s to state s' and the corresponding transition rate. The total transition rate out of state s is given as $q_s = \sum_{k \neq s} q_{s,k}$. The transition probabilities of the

embedded chain can be obtained as

$$\begin{aligned} p_{s,s'} &= \lim_{\Delta t \to 0} P[X_{t+\Delta t} = s' | X_{t+\Delta t} \neq s, X_t = s] \\ &= \lim_{\Delta t \to 0} \frac{P[X_{t+\Delta t} = s', X_{t+\Delta t} \neq s | X_t = s]}{P[X_{t+\Delta t} \neq s | X_t = s]} \\ &= \frac{q_{s,s'}}{\sum_{k \neq s} q_{s,k}}. \end{aligned} \tag{6.3}$$

6.3.2 Markov Chain-based Security Estimation Model

From the DTMC model formalized in the previous sub-section, the transition probability matrix of the state space $\mathcal{S}$ can be extracted, which can be presented as:

$$\mathcal{P}_{ss'} = \mathbb{P}[\mathcal{S}_{t+1} = s' | \mathcal{S}_t = s] \tag{6.4}$$

where s and s' represent the current and next states of the migration system. The resulting transition probability matrix is presented in Table 6.5, where the states are critical bandwidths specified in Table 6.4.

Table 6.5: Transition Probability Matrix

States	BW1	BW2	BW3	BW4	BW5	BW6
BW1	0.9698	0.0302	0	0	0	0
BW2	0.5276	0	0.4724	0	0	0
BW3	0	0.5227	0	0.4773	0	0
BW4	0	0	0.5191	0	0.4809	0
BW5	0	0	0	0.5134	0	0.4865
BW6	0	0	0	0	0.7659	0.2341

This estimation model is designed in the prospect of a security-associated response for the ω_R variation of the SMC. In fact, the maximum likelihood state is to be determined, after the current state in the $\mathcal{S}$. This can be achieved by determining the highest transition probability in the $\prod$ matrix formed by all possible $\mathcal{P}_{ss'}$ values and the response would be the suited SP that adheres to the situation. However, a Mapping (MAP) function is required for deciding the most suited SP for the ω_R of the estimated state s': $max[\mathcal{P}_{ss'}]_{s'=0}^{n} \rightarrow max[\prod_0^n] \rightarrow s'_{max} \rightarrow \omega_{R,s'_{max}} \rightarrow MAP(\omega_R) \rightarrow spi_s$. In fact,

the $MAP()$ function contrives a matrix of spi values that are suited for all possible ω_R in the SMC post-authentication. Quoting sub-section 6.2.3.3, most of the entries in the $MAP()$ matrix would represent the spi with the highest security setting. This estimation process can be integrated as an alternative to the SMC sensing function $extract(\omega_R)$ specified in Algorithm 1.

6.3.2.1 Validating the Proposed Estimation Model

A simulation was conducted to determine the completion times of different security schemes in which security estimation is utilized and not utilized. Out of the selected bandwidths in Table 6.4, ω_Rs of BW2, BW3, BW4, and BW5 were selected for this simulation for representing the critical states. For the sake of the simulation, four ω_R changes were assumed during the migration process in the SMC. The knowledge on transitions from Table 6.5 was utilized in the model when the $SWITCH$ function activates at the estimated ω_R changes. A 250MB MES was considered for this migration, under six different schemes of security application. The details of the $MAP()$ function and the results of this simulation are shown in Table 6.6. Accordingly, the results with the estimation model produce lesser T_Ms compared to both general and heavy security schemes.

6.4 Prototype Service Migration Security Framework

A prototype platform was implemented for evaluating the feasibility of the proposed security optimization methodology in a pragmatic context. Fig. 6.11 illustrates the architectural view of the developed prototype platform, its specifications, and functions. This section explicates the developed experimental platform and the conducted feasibility study.

Table 6.6: Specifications of the $MAP()$ function for selected Security Settings (SSs) and the Simulation Results

States	**BW Util.**	**SS-1**	**SS-2**
$MAP()$ ***function***			
BW2	87.5%	AES-256	AES-192
BW3	91%	AES-256	AES-192
BW4	93.75%	RC4	RC4
BW5	97%	RC4	RC4
Security Scheme	**Description**		T_M
Simulation Results			
Plaintext	Without any security and estimation		12.24 s
Light Security	RC4 without estimation		12.85 s
General Security	AES-128 without estimation		16.28 s
Heavy Security	AES-256 without estimation		18.36 s
Scheme 1	SS 1 with estimation		13.81 s
Scheme 2	SS 2 with estimation		13.61 s

6.4.1 Specifications of the Prototype Framework

A high-performance server (i.e. Processor: Intel Xeon 2.2 GHz 24 CPU, RAM: 98 GB, OS: Ubuntu 16.04 LTS) was launched as the gNB_S, while Processor: Intel Xeon 2.4 GHz 4 CPU, RAM: 8 GB, OS: Windows Server 2016 64bit specifications was inhibited by the server selected to emulate the gNB_R. The g2g SMC was developed with a Java socket-based Inter-Process Communication (IPC) approach, where channel interface S and channel interface R were launched as independent singular instances specific to the migration process. SP registry was maintained in files at both ends. The MESs were emulated as docker containers running on the virtualized platforms at both emulated gNBs.

The migration process is assumed to be initiated at the invoking of the migration handler by the channel interface S. Upon hibernation of the MES container, it will be compressed into a reduced size. Then the compressed instance is encoded using the Base64 format for convenience. The fragmenting process will then divide the information into an array of data fragments. The size of the data fragment is determined by the data handling capacity of the channel and utilized technology. The security handler is implemented as a separate program in which it will apply the security mechanisms (encryption,

integrity, timing, ...etc.) to each fragment and concatenate to form the sending message. The message upon receiving at the R interface will be subjected to the reverse procedure until the decompressed container image is formed.

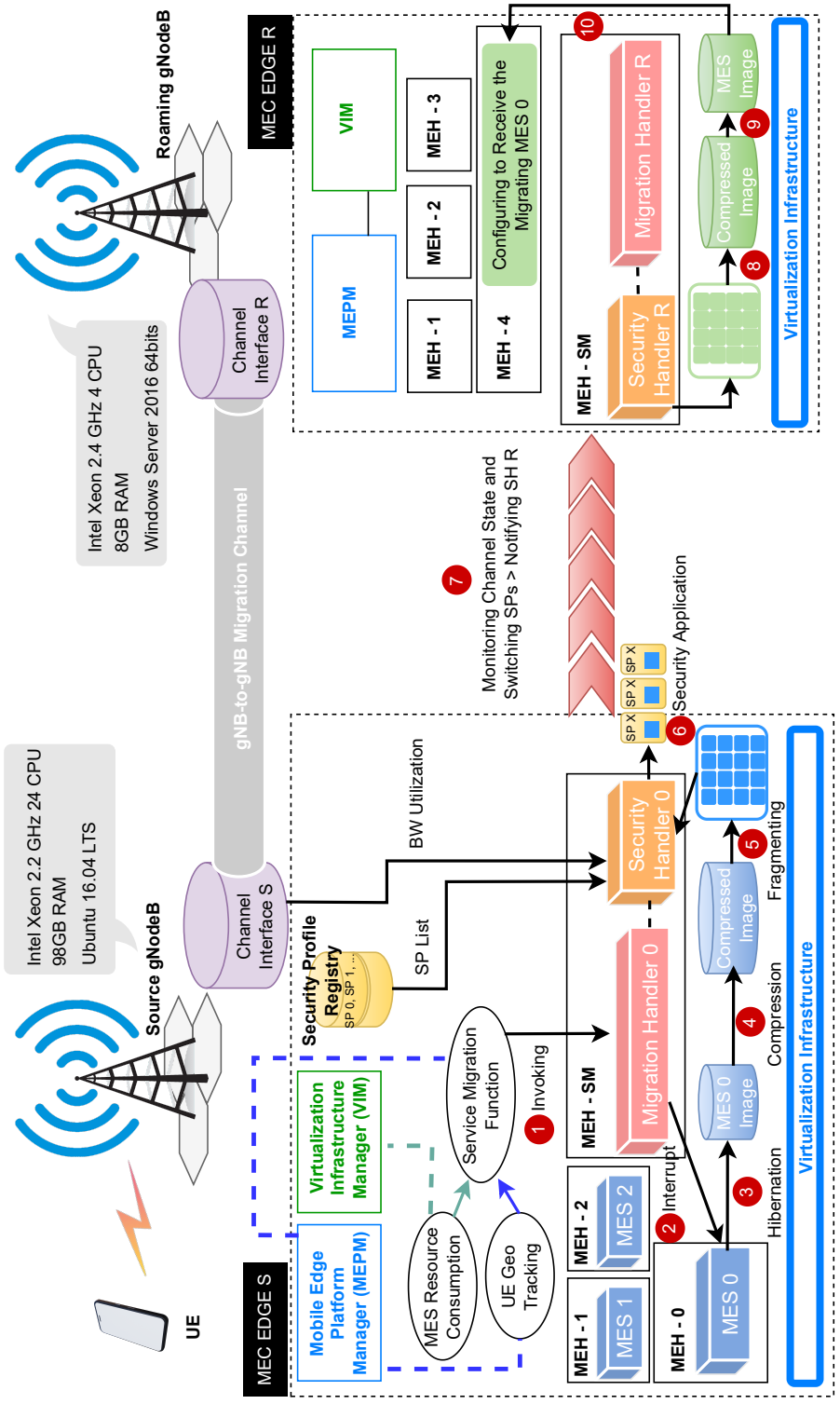


Figure 6.11: Implementation Setup of the Proposed SP Application and Migration Process from an Architectural Viewpoint

Table 6.7: Emulation Results of the Prototype for Different Security Setting (SS)/ SPs

SS **No.**	**Algorithms**	**Migration Delay** (t_M:ms)	**Decrypt. Delay** (t_D:ms)	**Complete Time** (t_C:ms)
LAYERS = 1				
001	AES-256	7247	8512	28966
002	AES-192	7072	8469	27799
003	AES-128	6746	8311	27303
004	BlowFish	1933	3442	17354
005	RC4	1060	3040	16260
100	AES-256+AES-128	6804 (6.11%)	8494 (0.2%)	27427 (5.31%)
200	AES-256+BF	4497 (37.94%)	6017 (29.31%)	22745 (21.48%)
300	AES-256+RC4	4152 (42.7%)	5669 (33.4%)	21681 (25.15%)
400	AES-256+AES-128+BF+RC4	4060 (43.97%)	5962 (29.95%)	22137 (23.58%)
LAYERS = 2				
1001	AES-256/SHA-512	7007 (7.1%)	9136 (-3.65%)	28050 (4.39%)
1002	AES-256/Kecc-512	7545	8814	29339
1003	RC4/SHA-1	1725 (77.14%)	3203 (63.66%)	17662 (39.8%)
1004	RC4/Kecc-256	3117 (58.7%)	3901 (55.74%)	23710 (19.19%)
1101	AES-256/SHA-512+AES-192/SHA-512+AES-128/SHA-512+RC4/SHA-512	5846 (22.52%)	7529 (14.58%)	27563 (6.05%)
1102	AES-256/SHA-1+AES-192/SHA-1+AES-128/SHA-1+RC4/SHA-1	5822 (22.83%)	7276 (17.45%)	26310 (10.32%)
1103	AES-256/SHA-512+AES-128/SHA-512+BF/SHA-512+RC4/SHA-512	4503 (40.32%)	6202 (29.63%)	24219 (17.45%)
1104	AES-256/SHA-1+AES-128/SHA-1+BF/SHA-1+RC4/SHA-1	4734 (37.26%)	6654 (24.5%)	24764 (15.6%)

6.4.2 Emulations

The experiment was conducted for different fragment sizes and SPs. Several SPs were defined that cover both single and double layers for proving this concept. During the process, compression (t_{cp}), encoding (t_{enc}), fragmenting (t_{fra}), migration (t_M), decryption (t_D), decoding (t_{dec}), decompression (t_{dcp}) delays, and the completion time (t_C) were recorded. The t_C constitutes the entire process from step 1 to step 10 explicated in Fig. 6.11. The average t_{cp}, t_{enc}, t_{dec}, and t_{dcp} times are 4628 ms, 311 ms, 664 ms, and 1304 ms respectively; and these values are common for all the employed SPs. Table 6.7 indicates the results of the conducted emulation where the performance of different Security Settings (SSs) was tested. In this experiment, the fragment size was kept at 1 MB (i.e. 1024 KB). A dockerized container image [201] with a size of 114 MB was migrated. The developed compression function achieved a compression ratio of 37.06%. According to the t_M of the SS #001 (i.e. highest security standard), the developed framework meets the current standards as $t_M \approx 7s$ even with AES-256 encryption, which is an improvement compared to the results in [152]. These timing-based Key Performance Indicators (KPIs) form the benchmarked baseline for the conducted emulations and the developed MEC-SMSF. In this case, however, migration delay t_M and completion time t_C specifies the prime KPIs for measuring a migration process. I selected the t_M as a KPI since other delays corresponding to t_C are based on the MEC infrastructure capability, while t_M is dependent on the security handler and the SMC occupancy.

Table 6.7 shows the performance of the individual encryption algorithms or ciphers where only a single layer is employed. The SS #001 with the highest cost values was considered as the baseline for bench-marking other SSs under single-layered instances. SS #005 features the least cost due to its stream cipher nature. The second section represents the combination of SPs. The migration is assumed to be occurring under 4 phases with equal bearing on the migration data amount in accordance with results obtained from subsection 6.3.1.2 using the CTMC model. With SSs #100, #200, and #300 a 50% bearing is assumed for each SP where the SP is switched once. The

SP is switched three times in SS #400 and has a 25% equal bearing on each SP. The improvement of all the cost parameters of each SS is displayed in the table. Accordingly, it is obvious that a significant improvement can be made through the proposed dynamic security concept that reaches up to even 25% improvement on t_C under a single layer deployment.

Under the double layer SSs, a second layer for integrity violation detection was embedded as a Message Authentication Code (MAC) measure using well-known hashing algorithms of SHA and Keccak. SS #1002 represent the highest t_C, and the comparisons were drawn accordingly for double-layer instances. Contrasting, SS #1001 to SS #1002, and SS #1003 to SS #1004, it can be deduced that SHA is more efficient than Keccak in terms of MAC detection. Similarly, from SS #1101 to SS #1104, various combinations were emulated to derive their effectiveness in a double-layer deployment. Thus, these results further reiterate the validity of the proposed concepts and methodologies for attaining dynamic and flexible security.

6.5 Chapter Discussion

The relationship between the level of applied security and its ensued latency due to aggregated processing times and overhead is a vital research area for emerging technologies. Following the MEC-enabled Autonomous Vehicles as the impending use case, this chapter exploits the stated relationship via a framework proposed to ensure secure service migrations between gNBs. In order to standardize and formulate the applied security in the context of mobile networks, the term SP was introduced and its applicability and formation were described extensively. The SP enables the formation of security mechanisms for security protocols to be standardized across different network domains and applications. In fact, SP paves the path for the quantification of security. A cost function was derived for assessing the cost of individual security mechanisms, and hence computing the security cost of a SP. This cost is used to classify different security levels in which a dynamic optimal security selection can be carried out during the service migration process. Further, an estimation model

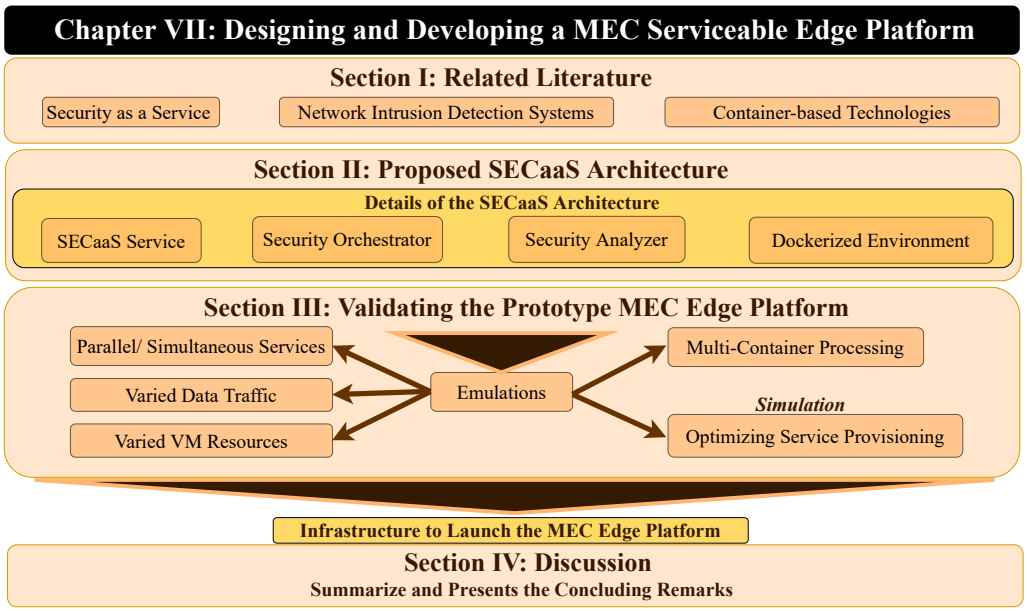
was introduced by adapting the Markov chains. The behavior of the formulated model was simulated. A prototype service migration security framework was implemented and its feasibility was inspected. This proof of concept will serve the emerging applications as a tool for managing security in critically dense traffic areas, where edge computing plays a key role in pragmatic deployments. Further, the concept of dynamic security is adding a confusing dimension to the overall security of the system, since the adversary has to contend with the uncertainty of the employed security mechanism(s). This confusion extends to all the requisites of security as confidentiality, integrity, and availability aspects.

CHAPTER

SEVEN

DESIGNING AND DEVELOPING A MEC SERVICEABLE EDGE PLATFORM

Extending the design presented in Section 2.2, this section presents the design and development details of a MEC serviceable platform that was intended to host a Security as a Service (SECaaS) deployment. Security as a Service (SECaaS) is an initiative for a service model that enables mobile and IoT consumers with diverse security functions such as Intrusion Detection and Prevention (IDPaaS), Authentication (AaaS), and Secure Transmission Channel (STCaaS) as a Service. The actual development of a MEC infrastructure is highly dependent on the integration of virtualization technologies to enable dynamic creation, deployment, and the detachment of virtualized entities that should feature interoperability to cater to the heterogeneous IoT devices and services. The intention of this research attempt was to identify the technologies that can host a realistic service in a MEC edge platform, where the assimilated knowledge would be essential for forming the prototype service migration framework MEC-SMSF and its future adoptions. Further, the proposed and developed design is validated for its performance based on its service level requirements (i.e. security service).



Chapter Organization

This short chapter is formed into four sections. Section 7.1 summarize the state-of-the-art literature related to the formation of SECaaS, NIDS, and container technologies. The proposed MEC serviceable SECaaS architecture is presented in Section 7.2, where its components are briefed for their function and purpose. The proposed architecture is validated through a prototype MEC edge platform in Section 7.3, following emulations conducted for simultaneous service operations, data traffic variation, VM resource variation, and Multi-container processing; while a simulation was formulated for optimizing the service provisioning based on the results gathered from the conducted emulations. Section 7.4 summarizes the importance of the developed MEC edge infrastructure.

7.1 Related Literature

Khettab et al. in [202] proposed an architecture that amalgamated Network Function Virtualization (NFV) and Software Defined Networks (SDN) to ensure 5G network slice security through security functions provisioned as Virtual Net-

work Functions (VNFs). The goal of the proposed model was to perform Optimal Resource Provisioning to Reduce the Operational Expenditure (OPEX) by launching VNFs within different slices to leverage the elasticity and flexibility of NFV. The security tools of Snort, Suricata, and Ntopng are launched as VNFs to form the SECaaS model that attributes dynamic deployment, performance tracking, and predictive auto-scaling. A performance evaluation was conducted to determine the scalability of the SECaaS tools. However, this paper does not explicitly specify the possibility to launch the proposed architecture in edge computing scenarios.

Boudi et al. in [203] conducted an assessment of container-based technologies to select the best approach for furnishing security mechanisms to resource-constrained edge nodes. Two case studies of Factory 4.0 and Smart Home were stated by the paper to realize the applicability of SECaaS-based edge services. Docker containerization was utilized for evaluating the performance with a Raspberry Pi 3 edge node tested for various scenarios.

Sforzin et al. in [204] proposed a robust and scalable security solution for IoT environments to defend against cyber attacks. In this research, an intrusion detection architecture was presented utilizing Raspberry Pi as the core commodity for simulating a resource-constrained IoT node. Snort was used to evaluate the performance of the simulated node with the intention of determining the optimal configuration for sustainable operation.

Tripathi et al. in [205] explored the possibility of utilizing Raspberry Pi as an Intrusion Detection System (IDS) against cyber attacks. In addition, the implemented security functions included a honeypot, packet analyzer, and firewall. The proposed system was tested in a home network with Snort as the respective IDS. The current literature fails to consider standardized MEC architecture for realizing their goals. Even though the stated related researchers conducted experiments on Snort, Suricata, and firewall functions, their performance in simultaneous operation sharing the same virtual resources was not considered. Thus, the relevant experiments in Section 7.3 were conducted to validate the proposed SECaaS architecture.

7.2 Proposed MEC-enabled SECaaS Architecture Design

As depicted in Fig. 7.1, the MEC edge level is the sole focus of this section. The MEC edge level is bound to serve several Mobile Edge Services (MESs) such as AR, video streaming, and V2E, in addition to the SECaaS services. Thus, each MES was to be provisioned under a Mobile Edge Host (MEH); which is the main operating entity in the MEC edge level [25]. MEHs are launching Mobile Edge Applications (ME Apps) related to a particular service that interface with a User Equipment Application (UE App) in the UE [41]. As these MEHs might be commissioned to handle a considerable amount of mobile or IoT devices (UE Apps), a function of a MEH could be managed by a VM to cater to the required resources. Though, ME Apps should attribute lightweight virtualization characteristics due to flexibility, less resource consumption, dynamic creation, and deletion requirements. Docker is the ideal technology that suits these criteria [203].

The principal proposal of this section is to employ docker containerization to launch multiple ME Apps as containers that are operating inside a VM. Mobile Edge Platform (MEP), the management entity within the MEH is to be launched as a content manager, or the manager node in the Docker swarm mode [206]. In that aspect, MEP should perform the functions of; i) creation and termination of containers according to UE App requests, ii) selection of services for created containers and auto-configuration, iii) Networking among containers, and iv) managing the container resources. The containerized network that connects all the containers and MEP together is outbound by a virtual gateway. This virtual gateway is connecting the internal container network to other VMs at the edge level and to the Internet with VM network adaptation.

The MEPM is developed as a VM while each MES offered by the MEC system is launched as a VM. The VIM functionality is to be launched using the current hypervisor technologies (i.e.: Microsoft Hyper-V or VMware ESXi). The entire MEC edge level is launched as a singular virtual platform governed by the selected VIM technology. Though, a cluster of servers is required to

host the virtual platform which should be dynamic. The most suited hypervisor technology should be evaluated through performance valuation considering their flexibility, resource utilization for creating and maintaining VMs, compatibility, interoperability, and support for heterogeneous mobile services. A bare metal hypervisor is an ideal launching platform for this proposed environment.

As illustrated in Fig. 7.1, each security service (i.e. ME App) offered to the consumer is executed as a Docker container. Operating a single ME App to cater to a specific UE App would raise issues in terms of scalability when a multitude of UEs are connected to the SECaaS MEH and requesting services simultaneously. Thus, the following approach is to launch ME Apps to serve more than one UE App and manage the service operations following the Service Function Chaining (SFC) concept.

7.2.1 Security Orchestrator (SO)

The service requests are handled by the SO acting as the MEP for this MEH. Once a service request is approved by the SO, it will create and configure a container with the approved service or utilize the services of an existing container. SO is monitoring the resource utilization of the Virtualization Infrastructure (VI) in the prospect of optimizing efficiency.

7.2.2 Security Analyzer (SA)

SA, operated as a ME App is acquiring and storing security-related statistics and credentials within the system. All the red flags drawn from the IDPaaS instances are gathered and conveyed to the SECaaS centralized server for updating their signature profiles and defense strategies. In addition, threat assessment and prediction constructs are executed at the SA with the gained insights from SECaaS centralized servers. The links to verify the user credentials (i.e. from an external database of authentication credentials) are contained in the SA. These links are viable for AaaS and STCaaS services. Moreover, performance statistics of all security services are recorded in SA.

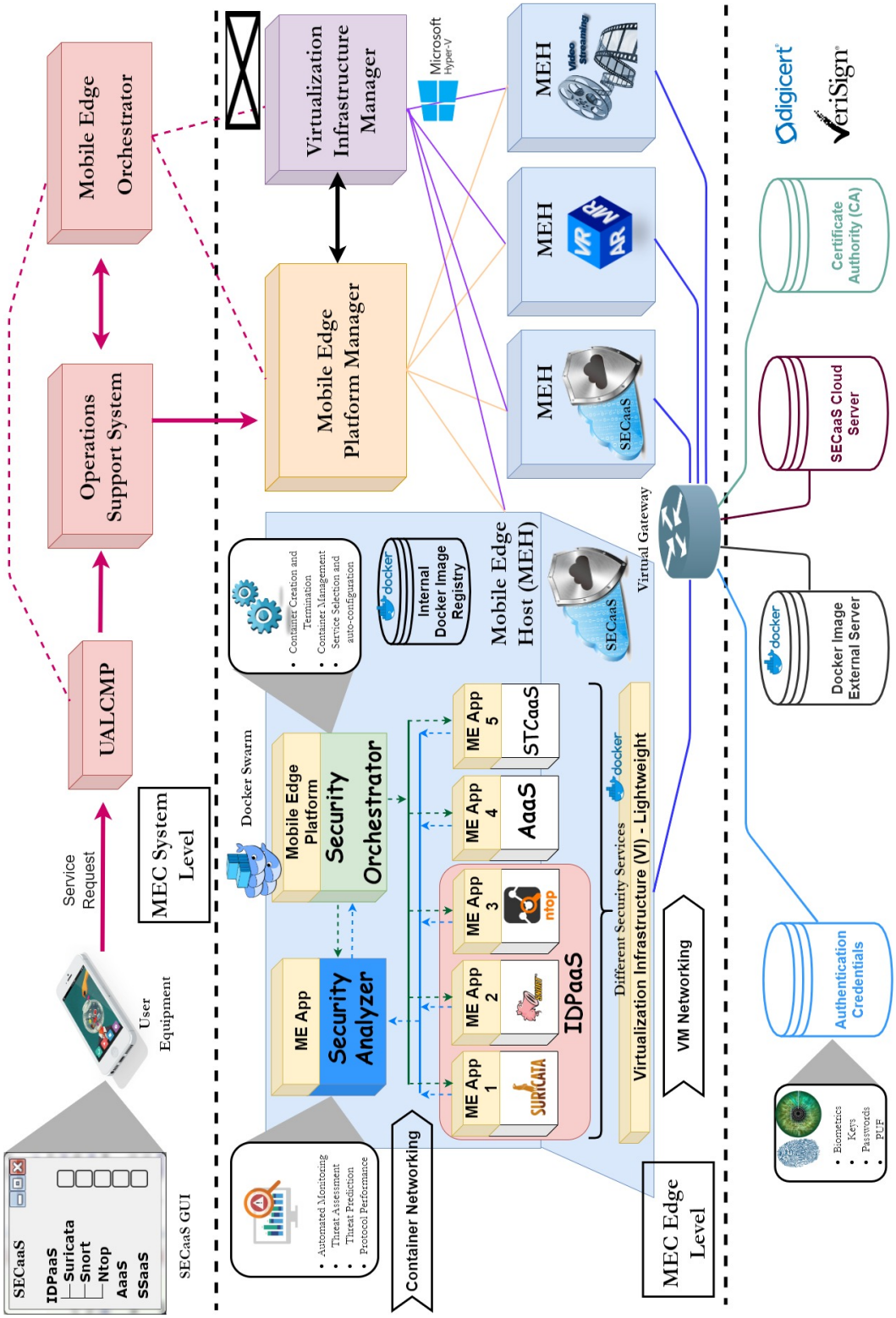


Figure 7.1: The Proposed SECaaS-based MEC Edge Platform

7.2.3 SECaaS Services

The intended SECaaS concept offers three distinct services to the MEC subscribers. They are; Intrusion Detection and Prevention as a Service (IDPaaS), Authentication as a Service (AaaS), and Secure Transmission Channel as a Service (STCaaS). Under IDPaaS, different well-known Intrusion Detection and Prevention Systems (IDPSs) are operated and offered to the consumer with their strengths and weaknesses, so that the user is capable of selecting the best service suited to their requirement. Currently, the experiments are focusing on IDPSs of Suricata and Snort [207, 208]. In the AaaS directive, MEC edge level SECaaS MES is handling the authentication of any application or service desired by the subscriber as a Trusted Third Party (TTP). Thus, cloud-based services are validated for the subscriber while user credentials are conveyed and verified to the cloud service by the SECaaS MES. In STCaaS, SECaaS is creating a secure tunnel between the UE and the third-party service provider facility at the edge or at a distant location. In this initiative, an entire security protocol is engaged in securing the communication channel.

7.2.4 Dockerized Environment

Each security service has its own dockerized image contained in the internal registry of the MEH docker environment. Updated images are conveyed to the internal registry from the external docker server (i.e. Docker hub). Pulling, running, and building docker images are automated within the SO function. Security services are granted a distinct TCP port number for identifying the service throughout the entire MEC platform. This was attained by performing the port forwarding feature of docker containers to the host VM.

7.3 Validating the Prototype MEC Edge Platform

Fig. 7.2 illustrates the testing environment emulated for a functioning MEH as a Ubuntu VM under VirtualBox 6.0 hypervisor. The host machine inherits the specifications of core i7 2.50 GHz CPU with 12 GB RAM as specified in Table

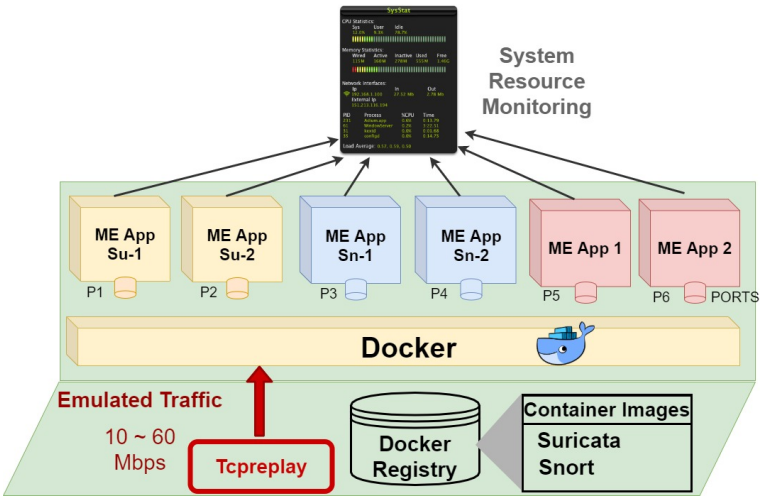


Figure 7.2: Prototype MEH Platform

7.1. Rules plays a vital role in an IDPS. The most important part comes when the user has to choose the number of rules that are required. The registered rules contain around 12,000 rules. The number of rules used here was kept as default for the registered rules. Since the goal was to test the performance of Snort and Suricata, rules were kept unchanged. Suricata 5.0.0 and Snort 2.9.15 with the registered rule set were employed. Both Snort and Suricata docker containers were tested for their performance with the network traffic streams emulated via tcpreplay 4.3.1. Mainly there are two parameters that are vital for security functions. They are the percentage of dropped packets processed by the IDS (denoted as d) and the percentage of alerts notified by the specific tool (denoted as a). In addition, CPU utilization (denoted as p) and RAM usage (denoted as r) were recorded to measure the performance of each container. The experiments were conducted in different scenarios. In order to emulate the tests, a pcap file called malware_exec.pcap with more than 800,000 packets was employed [209]. In this file 50% of the packets included malware content.

Table 7.1: Specifications and Configurations of the Prototype Testing Environment

Host PC Specifications

CPU	i7 2.50 GHz
RAM	12 GB
OS	Windows

VM Configurations

CPU Cores	**2 GB RAMs**
1	2
2	2
3	2
4	4

Table 7.2: Comparison of Suricata and Snort Performance in Simultaneous Operation

Factor	Suricata	Snort
CPU Utilization	9.48	81.10
RAM Usage	19.23	33.21
Packet Drops %	33.1	0
Alerts per Packets %	32.9	0

7.3.1 Parallel / Simultaneous Operation of Different Services

In order to validate the presented argument with regard to the parallel operation of docker container-based security instances, testing the simultaneous operation of Suricata and Snort is conspicuous. Thus, Table 7.2 tabulates the varied parameters of Suricata and Sort executed in the same environment. The observations suggest Suricata is performing more efficiently than Snort in terms of alerts. Thus, for the next testing scenarios, Suricata is being considered.

7.3.2 Varying Data Rate of the Traffic Stream for a Single Suricata Instance

Fig. 7.3 depicts the variation of drop percentage and alerts per total packets percentage. As expected d increases while a degrades with improving data

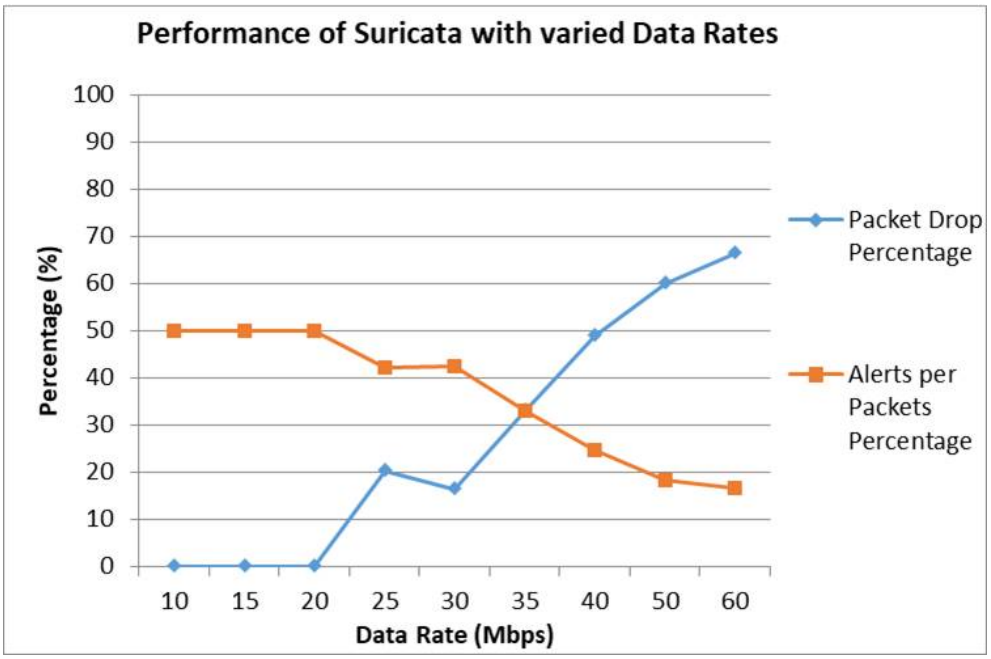


Figure 7.3: Suricata performance when traffic flow data rates are varying

rates due to the fact that the Suricata instance fails to read and process the traffic flow. According to the graph, 35 Mbps was observed to be the moderate value that could be considered for the experiments followed.

7.3.3 Variation of VM Resources for a Single Suricata Instance

Resources availability of the VM is vital for the outcome of IDS tools being used. Thus, the number of CPU cores and RAM allocated for the VM are tested for the configurations depicted in Table7.1. p and d values are alleviating with higher resources while a shows a minor increment. r values, however, don't vary significantly.

7.3.4 Multiple Container Processing

In Fig. 7.5, multiple Suricata containers were tested for determining their performance. The outputting alerts or a are increasing with each Suricata in-

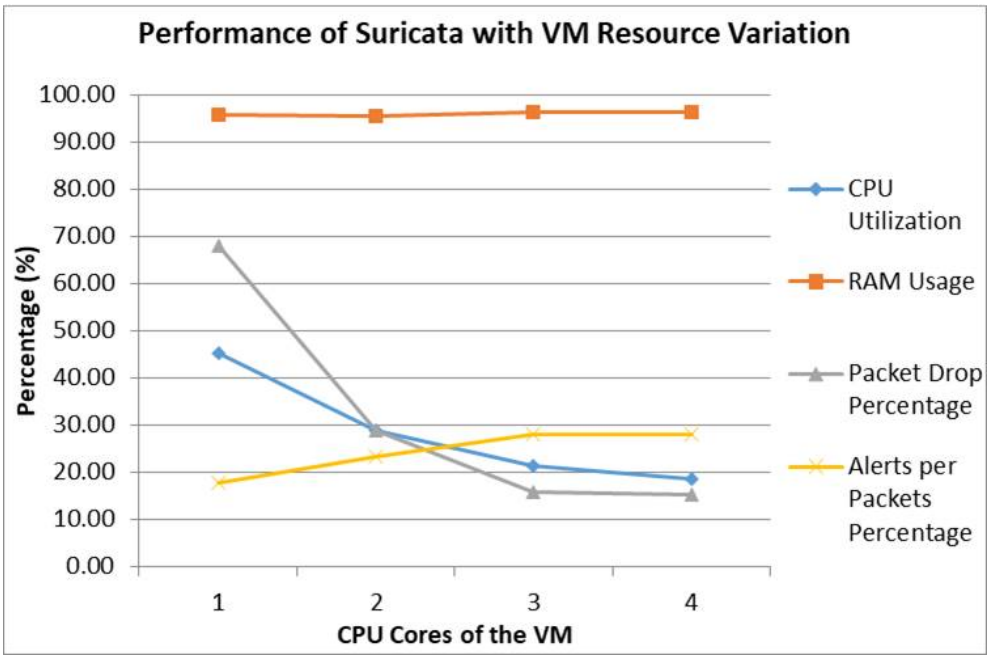


Figure 7.4: Suricata performance with varied VM resources

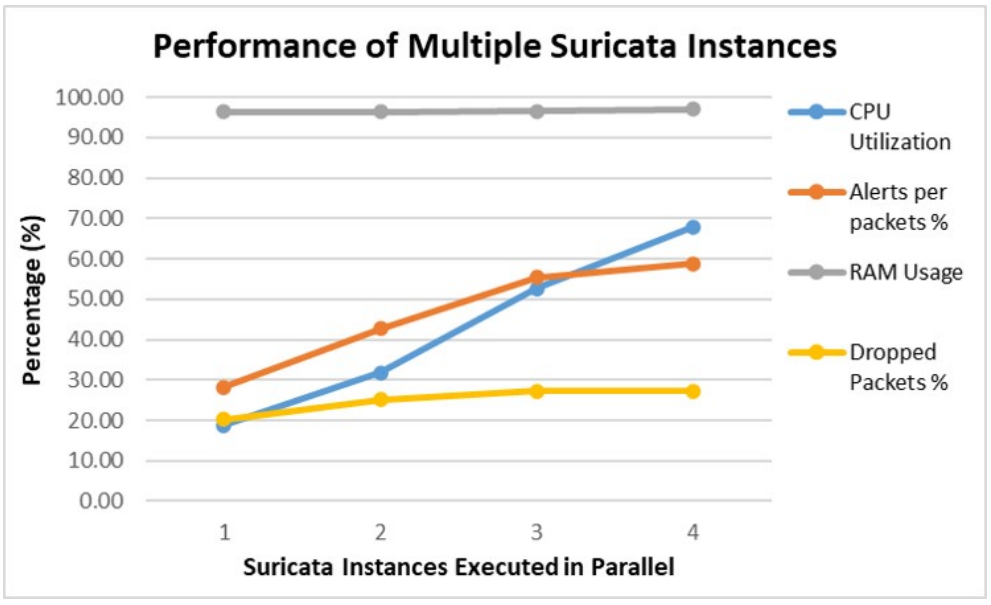


Figure 7.5: Performance of Suricata with multiple instances

stance operated in parallel. Though, the packet dropping percentage or d is stable between 25%-30%. The only drawback however is the accumulating

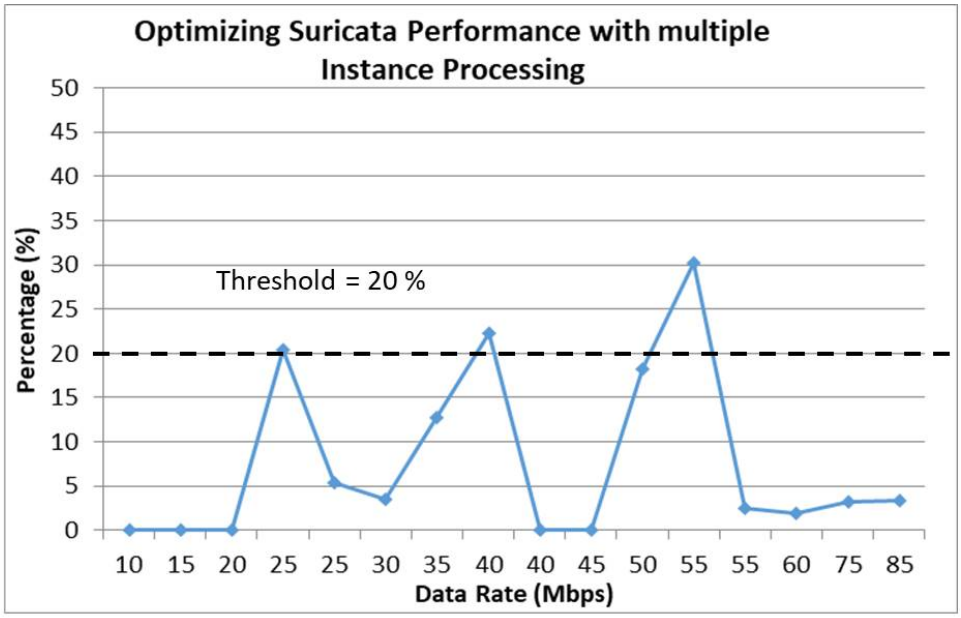
CPU usage with each Suricata process.

7.3.5 Optimizing Security Service Provisioning

According to the observations from Fig. 7.3, when the data rate is increasing, d is rapidly accumulating. One of the benefits of launching multiple security instances is its capability to handle higher data rates than isolated instances. Thus, Fig. 7.6 is formed by concatenating four different data sets extracted to measure d for different data rates when n number of Suricata instances are operating within a VM bearing n CPU cores. For this simulation, a threshold of 20% packet drop percentage was considered. Once the data rate goes beyond the 25 Mbps level, the threshold on d would be exceeded. Thus, such an instance is considered, where another Suricata container would be launched by the SO for balancing the load. Moreover, with dynamic resource allocation capabilities inherited by modern hypervisors, allocation of additional virtual CPU cores or expanding the virtual limits of the host CPU is plausible [210, 211]. Thus, this simulation is a feasible insight gained through this research that could be achievable via the SO functionality.

7.4 Discussion

According to the experiments conducted, parallel operation of multiple IDPSs as dockerized instances is plausible. Though, each IDPaaS tool is attributing different performance characteristics. Even the rules and signatures of each tool are differentiated in regard to robustness to various attacks. Thus, the number of alerts prompted by the tools is inconsistent for different pcap files. Though, Suricata detection accuracy is excelling more than its counterparts. Moreover, Suricata is capable of load distribution when multiple instances are operating. The simulation presented with the conducted experiments confirms the requirement for an orchestrating entity within each MEH. With the available technologies, it is possible to balance the IDPaaS-based network load among multiple security instances.



Suricata Instances	1	2	3	4
CPU Cores	1	2	3	4

Figure 7.6: Simulating packet drop optimization with multiple Suricata instances

The main goal of this chapter was to prove the feasibility of launching multiple security services simultaneously by employing virtualization technologies. The results and performance parameters of the developed SECaaS platform indicate the successful launching of an IDPaaS with Suricata and Snort tools where even multiple instances could also be operated simultaneously. Thus, the developed prototype MEC edge platform is successful in launching the specified security service. As an extension to the design presented in Section 2.2, this prototype implementation proves viable with this validation. Hence, the approach of utilizing both hypervisor-based and lightweight virtualization technologies can be followed to design the intended service migration framework for MEC. The methodologies and techniques adapted would be valuable for telecommunication, cloud, and security service providers to enhance their service models in order to cater to an extended consumer base with improved and guaranteed quality.

CHAPTER

EIGHT

DISCUSSION AND CONCLUSION

8.1 Discussion and Limitations

8.1.1 Thesis Discussion

It is obvious that edge computing paradigms are meant to elevate the current cloud-native serviceable infrastructures into a more efficient, secure, and privacy-preserving service that can overcome the intricacies of launching envisaged use cases such as autonomous vehicles, autonomous unmanned aerial vehicles, remote augmented reality based applications,...etc. Security of the edge computing paradigms, however, is not an area that was explored by the research community when this study started. Security being the main focus subject of the study, MEC was selected as the edge computing flavor for this Ph.D. due to its standardization and adaptation. ETSI, as the standardization body has moved from a conceptual Proof of Concept (PoC) in 2015 to a workable interactive environment in 2021 through ETSI MEC Sandbox [46]. In addition, there are various MEC-based test beds and environments developed to test its feasibility. Thus, MEC is leading to be adapted earlier than other edge computing paradigms in a global context. The standardized architecture presented by the ETSI is quite realistic, and current technologies can be integrated to develop the concept. More importantly, MEC standardization

focuses on the big picture of launching the impending technologies on a global scale laid on top of the prevailing mobile network infrastructure serviced by the cloud-native storage and processing environments. Therefore, the selection of MEC for this study was a logical decision.

This study was initiated as an attempt to investigate the security issues related to the MEC paradigm. In order to get a realistic idea of the existing security vulnerabilities, a pragmatic deployment context should be considered, especially for a technology such as MEC, where an industrial-level deployment is still not available. Assimilating the standardization (i.e., ETSI, NGMN, Huawei,...etc.) and preliminary research conducted on the subject, I managed to delineate a probable MEC deployment architecture (i.e., Fig. 3.1) to serve this purpose. This deployment architecture was further extended to envisage the deployments of emerging use cases enabled by MEC in [22, 17, 24]. The formalized deployment architecture was intrinsic to not only investigate the security vulnerabilities but to unsheathe the Ph.D. problem and to develop the MEC-SMSF demonstrated in this thesis.

The investigation conducted on revealing security vulnerabilities and attack vectors was quite thorough. And this investigation led to the selection of service migration as the focus of the Ph.D. study. Though the phenomenon of service migration was already available with cloud computing resource migrations, edge-to-edge migrations are unique to edge computing paradigms. It didn't attract as much attention in the research community as the offloading concept. Despite research work existed on improving the efficiency of service migrations, security was not a priority in the available literature. The investigations on the service migration phenomenon in the MEC context revealed various probable attack vectors that were perpetrated in a Man-in-the-Middle fashion, assuming edge infrastructures were uncompromised. Though instilling or injecting malicious content could infuse more impairment in contrast to traditional threats such as Replay, Relay, sniffing, spoofing, or impersonation attempts,....etc. A malicious agent penetrating into the service migration channel as a segment of the migration content could compromise the system by exploiting the weaknesses of the virtualization technologies (i.e., VM resource allocation) that enable autonomous operation. Once a malicious MES

is transferred and installed in the roaming gNB, it is arduous to detect the infection other than observing the anomalous behavior spanning a considerable time. The flexibility offered by 5G for local operators to launch their own gNBs is improving the confusion for the service migrations, as a malicious agent could easily exist within the operating range. This fact is prioritizing the authentication of operating MEC-enabled gNBs as a mandatory requirement. In addition, as MESs are virtualized service instances or containers, we can assume at some point in time, an intruder will be able to instill a malicious MES image into an image repository. If this attempt is successful, this violation cannot be detected by any standard security detecting system since the said MEC image is already circulating through the system, compromising the sub-systems on its path. Thus, a proper mechanism is a requirement to reveal such violations of virtualized entity utilization. These security requirements are considered in this thesis.

Despite migration security is a mandatory requirement, resulting overhead due to the applied security mechanisms is a critical burden in the context of service migrations. This overhead is obviously expanding the size of the migrating content. Different security mechanisms generate different overheads and induce varying processing times (i.e., encryption, decryption, hashing, encoding,...etc.). Since security is not a singular function, there is more than one security mechanism applied to the plaintext (i.e., migrating content prior to any encryption or encoding) for ensuring CIA requirements in addition to the defense mechanisms targeted on timing or DoS-type threats. Therefore, the actual size of the overhead generated as a result of the intended security mechanisms is a reasonable doubt. When the security processing time is aggregated to the delay created by the additional security overhead, a formidable latency is infusing this migration process. In a circumstance where the complete channel BW is vacant, this won't cause any issues as the maximum expected throughput is applicable. Though in an area where mobile traffic is significant, the expected throughput of the g2g channel reduces, and the additional latency caused by the security mechanisms becomes more severe, and could even disrupt the service if the migration is not concluded prior to UE reaching the new MEC domain.

In order to mitigate the security issues associated with service migrations, an authentication protocol is a logical solution. In fact, identity verification is the key to circumventing impersonation or masquerading attacks. The current literature lacks a security protocol proposed for a g2g channel, specifically in the context of service migrations. Thus, the proposed MEC-SMAP protocol is the pioneering work that targets authentication in a g2g context for service migrations. As a trusted third party is a requirement for successful multi-party authentication, an entity that offers Migration Authentication as a Service (MAaaS) was proposed. Further, the pointed out issue of the possibility that a virtualized entity can get compromised since the decoupling of the virtual entities from the underlying infrastructure, and these virtual entities become self-sufficient, self-governing, and unregulated by the infrastructure handlers due to the completely autonomous environment, can be solved with this authentication protocol by introducing a verification entity for virtualized instances, or MESs. Therefore, the MVR entity was introduced to maintain a database of registered MESs and has the ability to verify the legitimacy of the migrating MES during the operation of the MEC-SMAP. This MEC-SMAP protocol was designed with federated identity verification (i.e., the same identity is verified by different trusted parties located at distributed domains) approach, which requires communication between multiple parties even though the actual communication should happen between two entities. In addition to the standard CIA security assuring measures, perfect forward secrecy, and DoS/DDoS mitigation techniques were integrated into the protocol. Moreover, the migration master key K_M is generated in a way (i.e., $K_M = H[(r_1 \times r_2 mod N)||n_S 3||n_R 2] = H[(r'_1 \times r'_2)||n_S 3||n_R 2]$) that different parameters are employed by gNB_S and gNB_R to do the computation (i.e., r_1, r_2 by gNB_S and r'_1, r'_2 by gNB_R), which makes the recreation of the key improbable for any outside party. The credentials related to this computation are conveyed in different stages of the MEC-SMAP to reduce the probability of an intruder extracting the credentials. All these security measures and design choices make this protocol an intricate design. The conclusion of the MEC-SMAP establishes a secure g2g channel ready for migration with K_M acting as a session key for the migration. The extensive verification results and the

validation conducted for the feasibility of the protocol improves its value as a deployment option.

With the requirement to optimize the employed security level in the service migration channel during the migration, the concept of Security Profile or SP was introduced. More importantly, this concept introduces a way to standardize the security application based on their security level and the purpose of the security mechanism. As explicated in Section 6.2, K_M acts as the configuration key for the SP. Due to the different sequences (different security algorithms) and configurations (block cipher modes, padding schemes,...etc.) the SP is representing, it adds more confusion aspect to the security in addition to the secrecy of the K_M. The Normalization process of the K_M is adding more diffusion aspects to the security. Apart from the complexity it infuses toward security, the introduced security cost model allows estimating the overall migration delay expected by applying each SP. This, in fact, makes it possible to benchmark the SPs based on the estimated migration delay and enables the migration handler to switch to the most suited SP depending on the BW utilization (i.e., residual bandwidth ω_R) of the service migration channel. This is the concept of dynamic security introduced in this thesis that extended to the MEC-SMSM function.

Since SP switching can take a significant switching delay depending on the BW utilization of the channel, a probabilistic estimation scheme was proposed and validated utilizing Markov chains. This estimation allows for minimizing the switching cost and performing the security management as possible with a real-time BW sensing of the channel. In order to evaluate this proposed method, a prototype of the proposed MEC-SMSF was developed, and MEC-SMSM was integrated into it. The design of the system depicted and described in Section 4.3 introduces a working model of a migration framework that can be adapted for any MEC-based edge platform. The emulations conducted on this platform not only proved the MEC-SMSM concept as a proof of concept but resulted in an infrastructure to conduct migrations efficiently and securely.

8.1.1.1 Final Observations

The average convergence time for the developed MEC-SMAP protocol is $2047ms \approx 2s$. This value acting as the main KPI for the MEC-SMAP is denoted as $t_{C(AP)}$. For this result, we have employed RSA-4096 bit, AES-256, SHA-512, and P-256 ECDH along with the DoS puzzle of complexity four that has a solving time of $\approx 70ms$. These settings of the cryptographic algorithms have the highest level of security for the current standards. As we have stated in Sections 4.2 and 5.2.1, pre-migration decision-making on the migration decisions is possible, and it allows the MEC-SMAP to instigate prior to UE reaching the boundary of the coverage domain of the current gNB_S. The complete clearance on the time to the coverage boundary should consider the migration completion time as well.

Table 8.1: KPI Statistics of the MEC-SMSM process and MEC-SMAP

Performance Metrics/ KPI	**MEC-SMSM KPIs**	
	Migration Delay (t_M)	**Completion Time ($t_{C(SF)}$)**
LAYERS = 1		
Highest Security	$7247ms$	$28966ms$
Lowest Security	$1060ms$	$16260ms$
Maximum Improvement	43.97%	23.58%
Minimum Improvement	6.11%	5.31%
LAYERS = 2		
Highest Security	$7545ms$	$29339ms$
Lowest Security	$1725ms$	$17662ms$
Maximum Improvement	40.32%	17.45%
Minimum Improvement	22.52%	6.05%
MEC-SMAP KPI		
Convergence time ==> $t_{C(AP)}$	$2047ms$	

At the convergence of the MEC-SMAP, the migration process is invoked. One of the tasks of conducting emulations for the MEC-SMSM concept is to get a sense of the security cost of each tested SP to complete the $MAP()$ matrix. Thus, benchmarking the timing-based KPIs is important. Though the migration delay t_M and completion time $t_{C(SF)}$ are the main KPIs of the MEC-SMSF, they are reliant on the size of the migrating image $\mathcal{M}$. Further, the

size of the fragment size being transferred m_i is another contributing factor for t_M and $t_{C(SF)}$. The statistics of the selected KPIs for MEC-SMSM and MEC-SMAP are tabulated in Table 8.1.

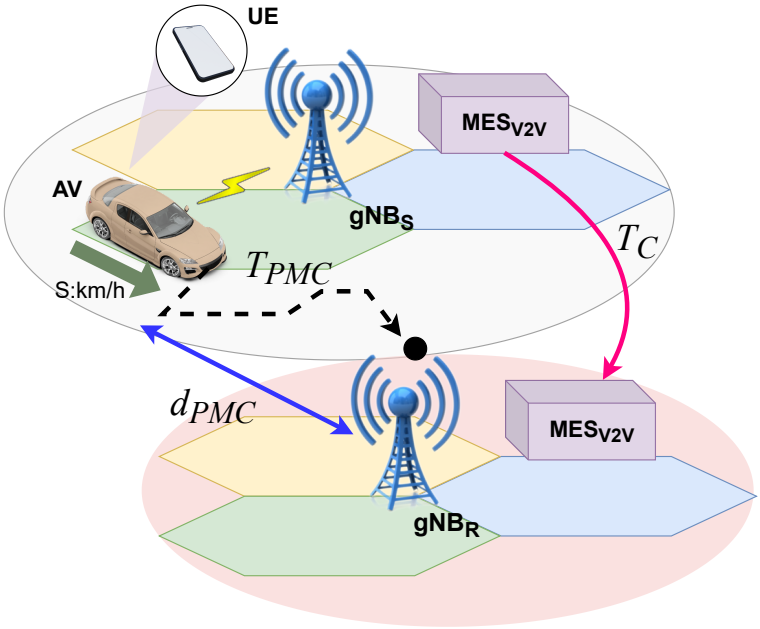


Figure 8.1: Pre-Migration Clearance

Despite that literature suggesting docker container startup times up to approximately $17s$ with multiple container deployments (i.e. 20 images) [212], singular containers can take up to $3.7s$ with Kata containers as specified in [213]. Though the development of the dockerized platform is exceeding the scope of this thesis and the container hibernation and startup processes are platform-specific operations that do not contribute to this comparison. Let denote clearance of the pre-migration as T_{PMC} (i.e. minimum time required to initiate the pre-migration process before UE reaches the next gNB domain). Fig. 8.1 indicates this pre-migration clearance in an illustrative context. According to these KPI results, we can deduce the T_{PMC}. If we assume the average UE speed as $50km/h$, the clearance on the distance (denoted as d_{PMC}) is approximated in Table 8.2. It can be seen that there is almost 41% improvement on $LAYER-1$ and 37.4% improvement on $LAYER-2$ of the clearance distance with higher and lower security settings.

Table 8.2: Deduced Pre-migration Clearance Statistics

T_{PMC}			d_{PMC}
$T_{PMC(L1,High-Security)}$	$31013ms$	$\approx 31s$	$430.6m$
$T_{PMC(L1,Low-Security)}$	$18307ms$	$\approx 18.3s$	$254.2m$
$T_{PMC(L2,High-Security)}$	$31386ms$	$\approx 31.4s$	$436.1m$
$T_{PMC(L2,Low-Security)}$	$19709ms$	$\approx 19.7s$	$273.6m$

8.1.2 Limitations

Since this research is conducted as a proof of concept for enhancing the security of service migrations in MEC environments, certain limitations exist in the design and the conducted validation or verifications. As the presented designs have a formidable potential, scalability and wide adaptability are major concerns at this prototype development stage. Thus, I am presenting the limitations of this thesis classified under the main contributions.

The MEC-SMSF implemented framework is equipping the MEC-SMAP and MEC-SMSM constructs separately. The complexity of the multiple connections maintained by gNB_S with various entities in the MEC-SMAP restricted the implementation from being integrated together.

8.1.2.1 Limitations in the proposed MEC-SMAP

In spite of the design of MEC-SMAP proposed for a 1-to-N authentication model, the implementation only followed a 1-to-1 approach. Thus, the current implementation mentioned in Appendix A.7 has the said limitation but can be extended to perform the 1-to-N model in a multi-threaded environment.

Synchronization issues: Synchronization is a clear limitation for any security protocol that utilizes timestamp-based Relay or Replay attack detection mechanisms. Synchronization issues will eventually stop the protocol initialization due to timestamp mismatching. This limitation will prompt the security handler to increase the current allowable clock skew. This is an opportunity for an intruder to capitalize on. This is much more severe for the proposed protocol as multiple entities are involved in the verification process, and even

a single re-transmission occurring due to a timestamp mismatch can compromise the protocol.

Where to Migrate?: This is a well-researched issue that many solutions exist already [145, 144, 214, 215, 199]. For this decision-making to be realistic, however, real-time resource utilization information on communication, operational, and backhaul capacities should be in possession by the gNB_S to make an accurate decision. This is quite challenging with highly scaled networks. Thus, a mechanism should be invented to extract the most critical information efficiently from potential gNB_Rs, and a limit on scalability should be determined through mathematical means.

Issues with the Authentication Model: MEC-SMAP is proposing a pre-migration authenticating process, where the gNB_S should be aware of all the surrounding potential gNB_Rs and their respective network credentials. The design suggests a 1-to-N authentication model that has to conduct multiple simultaneous authentications with selected N gNBs. Each gNB_R prompted with the authentication will have to reach both the TTP and the MVR entities, respectively. Though this sequence is practical for the limited value of the N, it is not highly scalable. Thus, an investigation has to be conducted to determine the optimum N for each migration instance.

Session Termination: In the proposed pre-migration authentication model, despite the authentication being conducted with several gNB_Rs simultaneously, only one gNB_R is selected at the end to perform the migration. Thus, all the other open sessions or channels should be terminated once the decision has been taken on where to migrate the service. This function can be easily embedded into the MEC-SMAP protocol in a practical context.

8.1.2.2 Limitation on the proposed MEC-SMSM Function

QoS and QoE of MES: Current SP selection process does not consider the priority status. In other words, QoS (QCI) or QoE indications of the MES

when selecting the respective SP. Even though the BW of the SMC is heavily occupied, high-priority service with priority QCI values still has to be secure enough, as their level of security cannot be reduced to the defined level. Thus, these factors should be considered in the SP decision-making process.

Modern Side Channel Dictionary Attack: Dictionary attacks are typically perpetrated to determine the key of a hash or a ciphertext by recording and storing either all possible ciphertexts or plaintexts. In this design, the adversary can observe the digest size of the migrating content and determine the employed algorithm through previous assimilation and proper recording. Then the intruder can target the instances with low-security algorithms and attempt to determine the credentials. Though this is a possible attempt, due to SP switching, not all the content is encrypted from the same algorithm. Due to the Normalization of the key, the exact key cannot be determined. Further, the faster switching between the SPs is limiting the possible test set for the intruder assimilation process.

Compatibility and Maintenence of SPs: In the current design, Security Handler maintains the SP handling functionality, and they are stored in the SP Registry, which is a simple database at the MEC gNBs. With the current formation, compatibility of the SPs is of concern. At the conclusion stage of the MEC-SMAP, all the relevant SPs are conveyed to the gNB_R as an array. I assume that gNB_R possesses all the security mechanisms specified by the sent SPs. Then, even if the sent SP is unavailable in the gNB_R registry, it can be formed with the available security mechanisms. But in reality, this might not be the case. Thus, a mechanism should be developed to obtain the respective security mechanisms from an external cloud environment. Though, this function exceeds the scope of this thesis, and can be visualized as an extension of the current work. The conveying of the SP array to gNB_R and its corresponding response to the gNB_S can be a threat vector of focus for adversaries. However, the MEC-SMAP encrypts this transfer with a key generated through PFS and additional methods explained above. This will ensure confidentiality, while integrity and availability are assured through the protocol

by its commonly implemented functions.

$MAP()$ **Function:** Current design of the $MAP()$ function is only considering Security cost as the main input. However, a measure of the security level should also be considered as an input parameter. This will enable the benchmarking of the SPs accurately.

Limited SP **Layers:** Despite the formalized SP model envisaging several layers of the specified SPs, the implementation of the MEC-SMSF incorporating the MEC-SMSM has only contended to 2 layers. Extending the implementation to a higher number of layers will improve the benchmarking process.

8.2 Conclusions and Future Work

8.2.1 Conclusion

This dissertation proposes a holistic solution for conducting *secure, dynamic, reliable, and efficient* service migrations for any use case or application that emerge with 5G and beyond technologies enabled by edge computing infrastructures. Though the designs formulated in this thesis focus on MEC, they can be extended to other edge computing paradigms by replacing the MEPM and VIM entities with the edge-level orchestrator and the hypervisor specified by the respective edge paradigm. From the facts presented in Chapter **??**, it is evident that the proposed design of the holistic MEC-SMSF is tamper-proof (i.e., secure) and facilitates a security management scheme that can ensure service efficiency to MEC subscribers in a reliable manner. The extensive validations and verifications to justify my claims in this thesis assure the reader of a verified genuine research attempt that can be followed trustworthily to improve their applications or use cases. This is early-stage research presented as a proof of concept to disseminate the ideals. Though there are limitations (i.e., as shown in Section 8.1.2) in the proposed design, it has great potential to overcome network performance-related issues in the current and emerging applications. Especially the concept of dynamic security embedding the

standardization of the security profile, where territorial or global compliance can ease service providers and consumers regarding security and trust for the services. To conclude this dissertation, the main highlights of this work are mentioned below.

- **A Map of Threat Vectors for a Typical MEC Deployment:** A holistic and thorough investigation conducted at the beginning of the Ph.D. resulted in an all-inclusive security analysis for MEC-based architectural deployments and MEC-enabled emerging use cases.
- **A Set of Solutions for MEC Vulnerabilities:** The same investigation has explored the existing security countermeasures and best practices to circumvent the revealed vulnerabilities.
- **A Design for an Efficient MEC Dynamic Serviceable Platform:** The investigations on the virtualization technologies have resulted in a design that can be adapted in actual MEC deployments. The presented validation of this design proves its feasibility.
- **Security Protocol for MEC gNB-to-gNB Communication:** Though the security protocol design was intended for service migration initiations and securing, it can be adapted for any communication between MEC-enabled mobile base stations.
- **Authentication for Virtual Entities:** As stated in Chapter 5, MEC-SMAP furnishes a verification scheme for virtualized services through the MVR. This novel approach proposed by the thesis can be enhanced in the future.
- **Dynamic Security Management Strategy:** The MEC-SMSM method demonstrates the possibility of optimizing the level of security and maintaining the service continuity of priority services under dense traffic areas.
- **MEC Service Migration Security Framework:** This work demonstrates the MEC-SMSF that successfully migrates service containers

from one gNB environment to another. These designed and implemented constructs can be utilized and even extended to develop real-world MEC-related deployments.

8.2.2 Future Work

The future work of this Ph.D. directive mainly focuses on the extension of the different functionalities proposed through the MEC-SMSF. Mainly, the MEC-SMAP security protocol and MEC-SMSM mechanism already available with the MEC-SMSF should be integrated together in an implementation context. The MEC-SMSF is expected to be extended into the MEC edge platform introduced in Chapter 7. This can be done through the previous work [29, 32, 30], which I was a part of. In addition, applying this concept to resource-constrained edge computing environments where IoT-based edge deployments are imminent would raise the value of dynamic security. Further planned future work is presented below.

8.2.2.1 MEC-SMAP

Blockchain for MES Virtual Entity Verification: The MES verification conducted by the MVR entity was implemented through a secure code generation mechanism. Though it is secure, its scalability and the trust issues with maintaining a large database with a MEC service provider should be investigated. Blockchain is a well-established trust verification system that can easily be integrated into the MES verification scheme.

Lightweight MEC-SMAP for Resource Constrained Applications: Though the MEC concept is typically deployed with an edge environment equipped with servers, certain IoT applications are scaling down the MEC edge deployment for resource-constrained edge nodes [27]. The high-security credentials employed by the MEC-SMAP are not deployable in these nodes. Thus, a lightweight security scheme with the same security level should be proposed for such applications.

8.2.2.2 MEC-SMSM

Security Quantification: For the $MAP()$ function, security quantification is a successful solution. With these approaches, the level of security or the measure of security will be introduced, which can be utilized to complete the dynamic security design of this thesis. The work conducted in [216, 217] improves the realization of this concept. However, integrating the quantification models for multi-layer security application schemes might be problematic.

Dynamic Security Estimation: The employed Markov chain-based security estimation model gives a solution for the delay caused by the SP switching process. In the current research scope, predictions or estimations are reached beyond the probabilistic models and utilize machine learning, reinforcement learning, or deep learning approaches. These methods can be adapted to improve the accuracy and decision time of the probable SP estimation process.

BIBLIOGRAPHY

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Pserspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[2] P. Beckman, C. Catlett, M. Ahmed, M. Alawad, L. Bai, P. Balaprakash, K. Barker, P. Beckman, R. Berry, A. Bhuyan *et al.*, "5g enabled energy innovation: Advanced wireless networks for science, workshop report," USDOE Office of Science (SC)(United States), Tech. Rep., 2020.

[3] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet," *ACM Computing Surveys (CSUR)*, vol. 52, no. 6, p. 125, 2019.

[4] R. Khan, P. Kumar, D. N. K. Jayakody, and M. Liyanage, "A survey on security and privacy of 5g technologies: Potential solutions, recent advancements and future directions," *IEEE Communications Surveys & Tutorials*, 2019.

[5] H. Hantouti, N. Benamar, T. Taleb, and A. Laghrissi, "Traffic Steering for Service Function Chaining," *IEEE Communications Surveys & Tutorials*, 2018.

[6] J. Costa-Requena, "Sdn integration in lte mobile backhaul networks," in *The International Conference on Information Networking 2014 (ICOIN2014)*. IEEE, 2014, pp. 264–269.

[7] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on Multi-Access Edge Computing for Internet of Things Realization," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2961–2991, 2018.

[8] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

[9] C. De Alwis, A. Kalla, Q.-V. Pham, P. Kumar, K. Dev, W.-J. Hwang, and M. Liyanage, "Survey on 6g frontiers: Trends, applications, requirements, technologies and future research," *IEEE Open Journal of the Communications Society*, 2021.

[10] M. Liyanage, I. Ahmad, A. B. Abro, A. Gurtov, and M. Ylianttila, *A comprehensive guide to 5G security*. Wiley Online Library, 2018.

[11] S. Singh, Y.-S. Jeong, and J. H. Park, "A survey on cloud computing security: Issues, threats, and solutions," *Journal of Network and Computer Applications*, vol. 75, pp. 200–222, 2016.

[12] K. Gai, M. Qiu, H. Zhao, and J. Xiong, "Privacy-aware adaptive data encryption strategy of big data in cloud computing," in *2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud)*. IEEE, 2016, pp. 273–278.

[13] M. Wang, P. P. Jayaraman, R. Ranjan, K. Mitra, M. Zhang, E. Li, S. Khan, M. Pathan, and D. Georgeakopoulos, "An Overview of Cloud based Content Delivery Networks: Research Dimensions and State-of-the-art," in *Transactions on Large-Scale Data-and Knowledge-Centered Systems XX*. Springer, 2015, pp. 131–158.

[14] A. Reznik, Y. Fang, and S. Ullah, "MEC in an Enterprise Setting : A Solution Outline," *ETSI White Paper #30*, vol. 2, no. 30, 2018, last accessed 16 May 2019. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp30_MEC_Enterprise_FINAL.pdf

[15] P. Ranaweera, V. N. Imrith, M. Liyanage, and A. D. Jurcut, "Security as a Service Platform Leveraging Multi-Access Edge Computing Infrastructure Provisions," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[16] P. Ranaweera, A. D. Jurcut, and M. Liyanage, "Survey on Multi-Access Edge Computing Security and Privacy," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1078–1124, 2021.

[17] P. Ranaweera, M. Liyanage, and A. D. Jurcut, "Novel MEC based Approaches for Smart Hospitals to Combat COVID-19 Pandemic," *IEEE Consumer Electronics Magazine*, vol. 10, no. 2, pp. 80–91, 2020.

[18] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile Edge Computing: A Survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.

[19] S.-C. Lin, Y. Zhang, C.-H. Hsu, M. Skach, M. E. Haque, L. Tang, and J. Mars, "The architectural implications of autonomous driving: Constraints and acceleration," in *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*, 2018, pp. 751–766.

[20] N. Jayaweera, N. Rajatheva, and M. Latva-aho, "Autonomous driving without a burden: View from outside with elevated lidar," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, 2019, pp. 1–7.

[21] R. Bruno and P. Ferreira, "Alma: Gc-assisted jvm live migration for java server applications," in *Proceedings of the 17th International Middleware Conference*, 2016, pp. 1–14.

[22] P. Ranaweera, A. Jurcut, and M. Liyanage, "MEC-enabled 5G Use Cases: A Survey on Security Vulnerabilities and Countermeasures," *ACM Computing Surveys (CSUR)*, vol. 54, no. 9, pp. 1–37, 2021.

[23] A. Jurcut, T. Niculcea, P. Ranaweera, and N.-A. Le-Khac, "Security Considerations for Internet of Things: A Survey," *SN Computer Science*, vol. 1, no. 4, pp. 1–19, 2020.

[24] P. Ranaweera, C. de Alwis, A. D. Jurcut, and M. Liyanage, "Realizing Contact-less Applications with Multi-Access Edge Computing," *ICT Express*, vol. 8, no. 4, pp. 575–587, 2022.

[25] P. Ranaweera, A. D. Jurcut, and M. Liyanage, "Realizing Multi-Access Edge Computing Feasibility: Security Perspective," in *2019 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2019, pp. 1–7.

[26] P. Ranaweera, A. Jurcut, and M. Liyanage, "Service Migration Authentication Protocol for MEC," in *2022 IEEE Global Communications (GLOBECOM) Conference*. IEEE, 2022.

[27] V. N. Imrith, P. Ranaweera, R. A. Jugurnauth, and M. Liyanage, "Dynamic Orchestration of Security Services at Fog Nodes for 5G IoT," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[28] V. N. Imrith, P. Ranaweera, S. Damree, and M. Liyanage, "Enabling Fog Computing based Dynamic Security Service Function Chaining for 5G IoT," in *2021 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, 2021, pp. 149–154.

[29] G. Dilanka, L. Viranga, R. Pamudith, T. D. Gamage, P. Ranaweera, I. A. Balapuwaduge, and M. Liyanage, "A Novel Server Selection Strategy for Multi-Access Edge Computing," in *2021 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, 2021, pp. 414–419.

[30] G. Dilanka, L. Viranga, R. Pamudith, T. Gamage, P. Ranaweera, I. A. Balapuwaduge, and M. Liyanage, "A Novel Request Handler Algorithm for Multi-Access Edge Computing Platforms in 5G," in *2022 IEEE 19th*

Annual Consumer Communications & Networking Conference (CCNC). IEEE, 2022, pp. 126–131.

[31] P. Ranaweera, A. D. Jurcut, and M. Liyanage, "Identifying Factors Enabling the Enhancement of Service Migration of Multi-Access Edge Computing," in *2022 Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit)*. IEEE, 2022.

[32] G. Dilanka, L. Viranga, R. Pamudith, T. D. Gamage, P. Ranaweera, I. A. Balapuwaduge, and M. Liyanage, "MEC-RHA: Demonstration of Novel Service Request Handling Algorithm for MEC," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2022, pp. 1–3.

[33] A. D. Jurcut, P. Ranaweera, and L. Xu, "Introduction to IoT Security," *IoT security: advances in authentication*, pp. 27–64, 2020.

[34] E. H. Jayatunga, P. S. Ranaweera, and I. A. M. Balapuwaduge, "Blockchain Advances and Security Practices in WSN, CRN, SDN, Opportunistic Mobile Networks, Delay Tolerant Networks," in *Revolutionary Applications of Blockchain-Enabled Privacy and Access Control*. IGI Global, 2021, pp. 1–34.

[35] V. Neerugatti and A. R. M. Reddy, "Secured Architecture for Internet of Things-Enabled Personalized Healthcare Systems," in *Internet of Things and Personalized Healthcare Systems*. Springer, 2019, pp. 75–80.

[36] R. Sathishkumar, C. Rani, and P. Ganeshkumar, "IoT based Monitoring of Container Vehicle for Secure and Reliable Delivery of Goods," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, 2018, pp. 628–633.

[37] R. Roman, J. Lopez, and M. Mambo, "Mobile Edge Computing, Fog et al.: A Survey and Analysis of Security Threats and Challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018.

[38] ETSI, “Mobile-Edge Computing–Introductory Technical White Paper,” *ETSI White Paper #1*, 2014, last accessed 16 May 2019. [Online]. Available: https://portal.etsi.org/Portals/0/TBpages/MEC/Docs/Mobile-edge_Computing_-_Introductory_Technical_White_Paper_V1%2018-09-14.pdf

[39] ETSI-GS-MEC, “Mobile-Edge Computing (MEC); Proof of Concept Framework,” *ETSI White Paper #5*, vol. 1, 2015, last accessed 16 May 2019. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/MEC-IEG/001_099/005/01.01.01_60/gs_MEC-IEG005v010101p.pdf

[40] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, “Mobile Edge Computing—-A Key Technology Towards 5G,” *ETSI White Paper #11*, vol. 11, no. 11, pp. 1–16, 2015, last accessed 16 May 2019. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf

[41] ETSI, “Mobile Edge Computing (MEC) Framework and Reference Architecture,” *ETSI White Paper #3*, 2016, last accessed 16 May 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/01.01.01_60/gs_MEC003v010101p.pdf

[42] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, P. Debashish, F. Jianping, F. Danny, G. Verin, W. Kuo-Wei, K. Kim, A. Rohit, O. Andy, L. M. Contreras, and S. Scarpina, “MEC in 5G Networks,” *ETSI White Paper #28*, vol. 28, no. 28, pp. 1–28, 2018, last accessed 16 May 2019. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp28_mec_in_5G_FINAL.pdf

[43] ETSI, “Multi-access Edge Computing (MEC); Phase 2: Use Cases and Requirements,” *ETSI White Paper*, vol. 2, 2018, last accessed 16 May 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/002/02.01.01_60/gs_MEC002v020101p.pdf

[44] ETSI-GS, “Multi-access Edge Computing (MEC); Framework and Reference Architecture,” *ETSI White Paper*, vol. 3, 2019, last accessed

16 May 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/02.01.01_60/gs_MEC003v020101p.pdf

[45] T. Bui, "Analysis of Docker Security," *arXiv preprint arXiv:1501.02967*, 2015.

[46] M. A. Hathibelagal, R. G. Garroppo, and G. Nencioni, "Experimental comparison of migration strategies for mec-assisted 5g-v2x applications," *Computer Communications*, vol. 197, pp. 1–11, 2023.

[47] R. Von Solms and J. Van Niekerk, "From Information Security to Cyber Security," *Elsevier Computers & Security Journal*, vol. 38, pp. 97–102, 2013.

[48] M. Mukherjee, L. Shu, and D. Wang, "Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018.

[49] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VM-based Cloudlets in Mobile Computing," *IEEE Pervasive Computing*, no. 4, pp. 14–23, 2009.

[50] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All One Needs to Know About Fog Computing and Related Edge Computing Paradigms: A Complete Survey," *Journal of Systems Architecture*, 2019.

[51] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in Mobile Cloud Computing: Taxonomy and Open Challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 369–392, 2014.

[52] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-art and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416–464, 2018.

[53] M. Liyanage, P. Porambage, A. Y. Ding, and A. Kalla, "Driving Forces for Multi-Access Edge Computing (MEC) IoT Integration in 5G," *ICT Express*, 2021.

[54] U. Shahid and R. Krenz, "Mobile Cloud Development with Software Defined 5G Networks using NFV (Network Function Virtualization Technologies)," *International Journal of Scientific & Engineering Research*, vol. 6, no. 9, pp. 1552–1555, 2015.

[55] D. Huang and H. Wu, *Mobile Cloud Computing: Foundations and Service Models*. Morgan Kaufmann, 2017.

[56] F. van Lingen, M. Yannuzzi, A. Jain, R. Irons-Mclean, O. Lluch, D. Carrera, J. L. Perez, A. Gutierrez, D. Montero, J. Marti *et al.*, "The Unavoidable Convergence of NFV, 5G, and Fog: A Model-Driven Approach to Bridge Cloud and Edge," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 28–35, 2017.

[57] F. B. Jemaa, G. Pujolle, and M. Pariente, "Cloudlet and NFV-based Carrier Wi-Fi Architecture for a Wider Range of Services," *Annals of Telecommunications*, vol. 71, no. 11-12, pp. 617–624, 2016.

[58] Y. Xu, V. Mahendran, and S. Radhakrishnan, "Towards SDN-based Fog Computing: MQTT Broker Virtualization for Effective and Reliable Delivery," in *8th International Conference on Communication Systems and Networks (COMSNETS)*. IEEE, 2016, pp. 1–6.

[59] L. Zhao, W. Sun, Y. Shi, and J. Liu, "Optimal Placement of Cloudlets for Access Delay Minimization in SDN-based Internet of Things Networks," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1334–1344, 2018.

[60] H. Xiang, W. Zhou, M. Daneshmand, and M. Peng, "Network Slicing in Fog Radio Access Networks: Issues and challenges," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 110–116, 2017.

[61] G. Gür, P. Porambage, and M. Liyanage, "Convergence of icn and mec for 5g: Opportunities and challenges," *IEEE Communications Standards Magazine*, vol. 4, no. 4, pp. 64–71, 2020.

[62] F. Song, Z.-Y. Ai, J.-J. Li, G. Pau, M. Collotta, I. You, and H.-K. Zhang, "Smart Collaborative Caching for Information-Centric IoT in Fog Computing," *Next Generation Wireless Technologies for Internet of Things*, vol. 17, no. 11, p. 2512, 2017.

[63] A. C. Baktir, A. Ozgovde, and C. Ersoy, "Enabling Service-Centric Networks for Cloudlets using SDN," in *IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2017, pp. 344–352.

[64] D. Schatz, R. Bashroush, and J. Wall, "Towards a More Representative Definition of Cyber Security," *Journal of Digital Forensics, Security and Law*, vol. 12, no. 2, p. 8, 2017.

[65] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2429–2453, 2018.

[66] A. Braeken, M. Liyanage, P. Kumar, and J. Murphy, "Novel 5G Authentication Protocol to Improve the Resistance Against Active Attacks and Malicious Serving Networks," *IEEE Access*, vol. 7, pp. 64 040–64 052, 2019.

[67] G. Yu, R. P. Liu, J. A. Zhang, and Y. J. Guo, "Tamperproof iot with blockchain," *arXiv preprint arXiv:2208.05109*, 2022.

[68] S. Alyahya, W. U. Khan, S. Ahmed, S. N. K. Marwat, and S. Habib, "Cyber secure framework for smart agriculture: Robust and tamper-resistant authentication scheme for iot devices," *Electronics*, vol. 11, no. 6, p. 963, 2022.

[69] M. Frustaci, P. Pace, and G. Aloi, "Securing the IoT World: Issues and Perspectives," in *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2017, pp. 246–251.

[70] L. Hathaway, "National policy on the use of the advanced encryption standard (aes) to protect national security systems and national security information," *National Security Agency*, vol. 23, 2003.

[71] 5GPPP-Security-WG, "5G PPP Phase 1 Security Landscape," *5GPP White Paper*, 2017, last accessed 11 November 2019. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP_White-Paper_Phase-1-Security-Landscape_June-2017.pdf

[72] Y. Jeon, H.-I. Ju, and S. Yoon, "Design of an LPWAN Communication Module based on Secure Element for Smart Parking Application," in *2018 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2018, pp. 1–2.

[73] L. Chen, L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone, *Report on Post-Quantum Cryptography*. US Department of Commerce, National Institute of Standards and Technology, 2016.

[74] T. Kumar, P. Porambage, I. Ahmad, M. Liyanage, E. Harjula, and M. Ylianttila, "Securing Gadget-Free Digital Services," *Computer*, vol. 51, no. 11, pp. 66–77, 2018.

[75] S. Gupta, B. L. Parne, and N. S. Chaudhari, "Security Vulnerabilities in Handover Authentication Mechanism of 5G Network," in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*. IEEE, 2018, pp. 369–374.

[76] P. Hao and X. Wang, "Integrating PHY Security Into NDN-IoT Networks By Exploiting MEC: Authentication Efficiency, Robustness, and Accuracy Enhancement," *arXiv preprint arXiv:1904.03283*, 2019.

[77] G. Choudhary, J. Kim, and V. Sharma, "Security of 5G-mobile backhaul networks: A survey," *arXiv preprint arXiv:1906.11427*, 2019.

[78] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, "Mobile-Edge Computing Architecture: The Role of MEC in the Internet of Things," *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 84–91, 2016.

[79] X. Zhang, A. Kunz, and S. Schröder, "Overview of 5g security in 3gpp," in *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2017, pp. 181–186.

[80] H. Zhu and C. Huang, "Availability-aware Mobile Edge Application Placement in 5G Networks," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[81] V. Jain, V. Laxmi, M. S. Gaur, and M. Mosbah, "ETGuard: Detecting D2D Attacks using Wireless Evil Twins," *Computers & Security*, vol. 83, pp. 389–405, 2019.

[82] M. H. Mahalat, S. Saha, A. Mondal, and B. Sen, "A PUF based Light Weight Protocol for Secure WiFi Authentication of IoT devices," in *2018 8th International Symposium on Embedded Computing and System Design (ISED)*. IEEE, 2018, pp. 183–187.

[83] G. Li and P. Bours, "Studying WiFi and Accelerometer Data Based Authentication Method on Mobile Phones," in *Proceedings of the 2018 2nd International Conference on Biometric Engineering and Applications*. ACM, 2018, pp. 18–23.

[84] Z. Zhao, G. Min, Y. Pang, W. Gao, and J. Lv, "Towards Fast and Reliable WiFi Authentication by Utilizing Visible Light Diversity," in *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2019, pp. 1–9.

[85] Y. Zhang, F. Ren, A. Wu, T. Zhang, J. Cao, and D. Zheng, "Certificateless Multi-Party Authenticated Encryption for NB-IoT Terminals in 5G Networks," *IEEE Access*, vol. 7, pp. 114 721–114 730, 2019.

[86] P. Gope, R. Amin, S. H. Islam, N. Kumar, and V. K. Bhalla, "Lightweight and Privacy-preserving RFID Authentication Scheme for Distributed IoT Infrastructure with Secure Localization Services for Smart City Environment," *Future Generation Computer Systems*, vol. 83, pp. 629–637, 2018.

[87] S.-C. Cha, M.-S. Chuang, K.-H. Yeh, Z.-J. Huang, and C. Su, "A User-friendly Privacy Framework for Users to Achieve Consents with Nearby BLE Devices," *IEEE Access*, vol. 6, pp. 20 779–20 787, 2018.

[88] P. Yu, J. Cao, M. Ma, H. Li, B. Niu, and F. Li, "Quantum-Resistance Authentication and Data Transmission Scheme for NB-IoT in 3GPP 5G Networks," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2019, pp. 1–7.

[89] S. Lal, T. Taleb, and A. Dutta, "NFV: Security Threats and Best Practices," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 211–217, 2017.

[90] D. Fang, Y. Qian, and R. Q. Hu, "Security for 5G Mobile Wireless Networks," *IEEE Access*, vol. 6, pp. 4850–4874, 2018.

[91] B. Jaeger, "Security Orchestrator: Introducing a Security Orchestrator In the Context of The Etsi NFV Reference Architecture," in *Trustcom/BigDataSE/ISPA, 2015 IEEE*, vol. 1. IEEE, 2015, pp. 1255–1260.

[92] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure Attribute-based Signature Scheme with Multiple Authorities for Blockchain in Electronic Health Records Systems," *IEEE access*, vol. 6, pp. 11 676–11 686, 2018.

[93] J. Kim, D. Kim, and S. Choi, "3GPP SA2 Architecture and Functions for 5G Mobile Communication System," *ICT Express*, vol. 3, no. 1, pp. 1–8, 2017.

[94] S. Chen, R. Ma, H.-H. Chen, H. Zhang, W. Meng, and J. Liu, "Machine-to-Machine Communications in Ultra-Dense Networks—-A Survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1478–1503, 2017.

[95] X. Lu, D. Niyato, N. Privault, H. Jiang, and P. Wang, "Managing Physical Layer Security in Wireless Cellular Networks: A Cyber Insurance Approach," *IEEE Journal on Selected Areas in Communications*, 2018.

[96] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C.-H. Hsu, "Edge Server Placement in Mobile Edge Computing," *Journal of Parallel and Distributed Computing*, 2018.

[97] C.-L. Chen, M.-L. Chiang, H.-C. Hsieh, C.-C. Liu, and Y.-Y. Deng, "A Lightweight Mutual Authentication with Wearable Device in Location-Based Mobile Edge Computing," *WIRELESS PERSONAL COMMUNICATIONS*, 2020.

[98] O. N. Hamoud, T. Kenaza, and Y. Challal, "Security in Device-to-Device Communications: A Survey," *IET Networks*, vol. 7, no. 1, pp. 14–22, 2017.

[99] K. Peng, V. Leung, X. Xu, L. Zheng, J. Wang, and Q. Huang, "A Survey on Mobile Edge Computing: Focusing on Service Adoption and Provision," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[100] H. Gupta, S. Mondal, R. Majumdar, N. S. Ghosh, S. S. Khan, N. E. Kwanyu, and V. P. Mishra, "Impact of Side Channel Attack in Information Security," in *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*. IEEE, 2019, pp. 291–295.

[101] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[102] A. Rahman, E. Hassanain, and M. S. Hossain, "Towards a Secure Mobile Edge Computing Framework for Hajj," *IEEE Access*, vol. 5, pp. 11 768–11 781, 2017.

[103] H.-M. Wang, T.-X. Zheng, J. Yuan, D. Towsley, and M. H. Lee, "Physical Layer Security in Heterogeneous Cellular Networks," *IEEE Transactions on Communications*, vol. 64, no. 3, pp. 1204–1219, 2016.

[104] K. Xiao, W. Li, M. Kadoch, and C. Li, "On the Secrecy Capacity of 5G MmWave Small Cell Networks," *IEEE Wireless Communications*, vol. 25, no. 4, pp. 47–51, 2018.

[105] P. Hao, X. Wang, and W. Shen, "A Collaborative PHY-Aided Technique for End-to-End IoT Device Authentication," *IEEE Access*, vol. 6, pp. 42 279–42 293, 2018.

[106] C. Marchand, L. Bossuet, U. Mureddu, N. Bochard, A. Cherkaoui, and V. Fischer, "Implementation and Characterization of a Physical Unclonable Function for IoT: A Case Study with the TERO-PUF," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 1, pp. 97–109, 2018.

[107] J. Zhang, X. Tan, X. Wang, A. Yan, and Z. Qin, "T2FA: Transparent Two-factor Authentication," *IEEE Access*, vol. 6, pp. 32 677–32 686, 2018.

[108] N. Islam, S. Das, and Y. Chen, "On-device Mobile Phone Security Exploits Machine Learning," *IEEE Pervasive Computing*, no. 2, pp. 92–96, 2017.

[109] B. Krupp, N. Sridhar, and W. Zhao, "SPE: Security and Privacy Enhancement Framework for Mobile Devices," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 4, pp. 433–446, 2017.

[110] B. S. Ainapure, D. Shah, and A. A. Rao, "Understanding Perception of Cache-based Side-Channel Attack on Cloud Environment," in *Progress in intelligent computing techniques: Theory, practice, and applications*. Springer, 2018, pp. 9–21.

[111] Y. Ai, M. Peng, and K. Zhang, "Edge Computing Technologies for Internet of Things: a Primer," *Digital Communications and Networks*, vol. 4, no. 2, pp. 77–86, 2018.

[112] S. Dalal and S. Devi, "Security Framework Against Denial of Service Attacks in Wireless Mesh Network Networks," *International Journal of Computer Networks and Communications Security*, vol. 4, no. 8, p. 237, 2016.

[113] H. Ko, K. Lim, J. Oh, and J.-K. K. Rhee, "Informatic Analysis for Hidden Pulse Attack Exploiting Spectral Characteristics of Optics in Plug-and-Play Quantum Key Distribution System," *Quantum Information Processing*, vol. 15, no. 10, pp. 4265–4282, 2016.

[114] E. Gündüzhan and K. D. Brown, "Narrowband Satellite Communications: Challenges and Emerging Solutions," *John Hopkins APL technical Digest, 2015*, vol. 33, 2015.

[115] ETSI, "Network Functions Virtualisation (NFV) Security: Report on Use Cases and Technical Approaches for Multi-layer Host Administration," *ETSI White Paper*, 2015, last accessed 16 May 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/009/01.01.01_60/gs_nfv-sec009v010101p.pdf

[116] ETSI-NFV-ISG, "Network Functions Virtualisation (NFV) Security: Cataloguing Security Features in Management Software," *ETSI White Paper*, 2015, last accessed 16 May 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/002/01.01.01_60/gs_NFV-SEC002v010101p.pdf

[117] T. Garfinkel, M. Rosenblum *et al.*, "A Virtual Machine Introspection Based Architecture for Intrusion Detection," in *Network and Distributed System Security Symposium*, vol. 3, no. 2003, 2003, pp. 191–206.

[118] I. Farris, J. B. Bernabé, N. Toumi, D. Garcia-Carrillo, T. Taleb, A. Skarmeta, and B. Sahlin, "Towards Provisioning of SDN/NFV-based

Security Enablers for Integrated Protection of IoT Systems," in *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2017, pp. 169–174.

[119] Z. Hu and Y. Yin, "A Framework for Security on Demand," in *13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2017, pp. 378–383.

[120] R. Solozabal, A. Sanchoyerto, E. Atxutegi, B. Blanco, J. O. Fajardo, and F. Liberal, "Exploitation of Mobile Edge Computing in 5G Distributed Mission-Critical Push-to-Talk Service Deployment," *IEEE Access*, vol. 6, pp. 37 665–37 675, 2018.

[121] X. Costa-Perez, A. Garcia-Saavedra, X. Li, T. Deiss, O. Delgado, A. Di Giglio, A. Mourad *et al.*, "5G-Crosshaul: an SDN/NFV Integrated Fronthaul/backhaul Transport Network Architecture," *IEEE Wireless Communications*, 2017.

[122] T. X. Tran, A. Hajisami, P. Pandealjhuaniy, and D. Pompili, "Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 54–61, April 2017.

[123] European Telecommunications Standards Institute. Available:https://www.etsi.org/. [Online; accessed March 18, 2018].

[124] M. Liyanage, I. Ahmad, M. Ylianttila, A. Gurtov, A. B. Abro, and E. M. de Oca, "Leveraging LTE Security with SDN and NFV," in *2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS)*. IEEE, 2015, pp. 220–225.

[125] M. Liyanage, A. B. Abro, M. Ylianttila, and A. Gurtov, "Opportunities and Challenges of Software-Defined Mobile Networks in Network Security," *IEEE Security & Privacy*, vol. 14, no. 4, pp. 34–44, 2016.

[126] 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, 2015.

[127] K. Leach, F. Zhang, and W. Weimer, "Scotch: Combining Software Guard Extensions and System Management Mode to Monitor Cloud Resource usage," in *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 2017, pp. 403–424.

[128] R. Wojtczuk, "Poacher Turned Gamekeeper: Lessons Learned from Eight Years of Breaking Hypervisors," *Black Hat USA*, 2014.

[129] A. Aljuhani and T. Alharbi, "Virtualized Network Functions Security Attacks and Vulnerabilities," in *7th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2017, pp. 1–4.

[130] Z. Yan, P. Zhang, and A. V. Vasilakos, "A Security and Trust Framework for Virtualized Networks and Software-Defined Networking," *Security and Communication Networks*, vol. 9, no. 16, pp. 3059–3069, 2016.

[131] M. Pattaranantakul, Y. Tseng, R. He, Z. Zhang, and A. Meddahi, "A First Step Towards Security Extension for NFV Orchestrator," in *Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*. ACM, 2017, pp. 25–30.

[132] M. Pattaranantakul, R. He, A. Meddahi, and Z. Zhang, "SecMANO: Towards Network Functions Virtualization (NFV) Based Security Management and Oorchestration," in *Trustcom/BigDataSE/I SPA, 2016 IEEE*. IEEE, 2016, pp. 598–605.

[133] L. R. Battula, "Network Security Function Virtualization (NSFV) Towards Cloud Computing With NFV Over Openflow Infrastructure: Challenges and Novel Approaches," in *Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on*. IEEE, 2014, pp. 1622–1628.

[134] C. Lorenz, D. Hock, J. Scherer, R. Durner, W. Kellerer, S. Gebert, N. Gray, T. Zinner, and P. Tran-Gia, "An SDN/NFV-Enabled Enterprise Network Architecture Offering Fine-Grained Security Policy Enforcement." *IEEE communications magazine*, vol. 55, no. 3, pp. 217–223, 2017.

[135] C. Basile, A. Lioy, C. Pitscheider, F. Valenza, and M. Vallini, "A Novel Approach for Integrating Security Policy Enforcement With Dynamic Network virtualization," in *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*. IEEE, 2015, pp. 1–5.

[136] E. Felstaine, O. Hermoni, and N. Sandlerman, "System, Method, and Computer Program for Managing Security In a Network Function Virtualization (NFV) Based Communication Network," Oct. 4 2016, uS Patent 9,460,286.

[137] S. Wang, J. Xu, N. Zhang, and Y. Liu, "A Survey on Service Migration in Mobile Edge Computing," *IEEE Access*, vol. 6, pp. 23 511–23 528, 2018.

[138] F. Zhang, G. Liu, X. Fu, and R. Yahyapour, "A Survey on Virtual Machine Migration: Challenges, Techniques, and Open Issues," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1206–1243, 2018.

[139] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, "Live Service Migration in Mobile Edge Clouds," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 140–147, 2018.

[140] M. Liyanage, J. Salo, A. Braeken, T. Kumar, S. Seneviratne, and M. Ylianttila, "5G Privacy: Scenarios and Solutions," in *2018 IEEE 5G World Forum (5GWF)*. IEEE, 2018, pp. 197–203.

[141] T. He, E. N. Ciftcioglu, S. Wang, and K. S. Chan, "Location Privacy in Mobile Edge Clouds: A Chaff-based Approach," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2625–2636, 2017.

[142] X. He, J. Liu, R. Jin, and H. Dai, "Privacy-aware Offloading in Mobile-Edge Computing," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[143] X. He, R. Jin, and H. Dai, "Deep PDS-Learning for Privacy-Aware Offloading in MEC-Enabled IoT," *IEEE Internet of Things Journal*, 2018.

[144] A. Nadembega, A. Hafid, and T. Taleb, "A destination and mobility path prediction scheme for mobile networks," *IEEE transactions on vehicular technology*, vol. 64, no. 6, pp. 2577–2590, 2014.

[145] A. Nadembega, A. S. Hafid, and R. Brisebois, "Mobility prediction model-based service migration procedure for follow me cloud to support qos and qoe," in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–6.

[146] J. Plachy, Z. Becvar, and E. C. Strinati, "Dynamic resource allocation exploiting mobility prediction in mobile edge computing," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2016, pp. 1–6.

[147] M. Mamman, Z. M. Hanapi, A. Abdullah, and A. Muhammed, "Quality of service class identifier (qci) radio resource allocation algorithm for lte downlink," *PloS one*, vol. 14, no. 1, p. e0210310, 2019.

[148] Eir. Eir (ireland) cell tower map. [Accessed November 30, 2020]. [Online]. Available: https://www.cellmapper.net/

[149] E. Access, "Further advancements for e-utra physical layer aspects," *3GPP Technical Specification TR*, vol. 36, p. V2, 2010.

[150] J. Ren, Y. He, G. Huang, G. Yu, Y. Cai, and Z. Zhang, "An Edge-computing based Architecture for Mobile Augmented reality," *IEEE Network*, vol. 33, no. 4, pp. 162–169, 2019.

[151] X. Zhang, W. Wu, S. Yang, and X. Wang, "Falcon: a blockchain-based edge service migration framework in mec," *Mobile Information Systems*, vol. 2020, 2020.

[152] L. Ma, S. Yi, and Q. Li, "Efficient service handoff across edge servers via docker container migration," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, 2017, pp. 1–13.

[153] A. A. Majeed, P. Kilpatrick, I. Spence, and B. Varghese, "Modelling fog offloading performance," in *2020 IEEE 4th International Conference on Fog and Edge Computing (ICFEC)*. IEEE, 2020, pp. 29–38.

[154] I. Miell and A. Sayers, *Docker in practice*. Simon and Schuster, 2019.

[155] Y. Zhang, R. H. Deng, E. Bertino, and D. Zheng, "Robust and universal seamless handover authentication in 5g hetnets," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 858–874, 2019.

[156] G. Karthick, G. Mapp, F. Kammueller, and M. Aiash, "Formalization and analysis of a resource allocation security protocol for secure service migration," in *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*. IEEE, 2018, pp. 207–212.

[157] M. Cui, H. Zhang, Y. Huang, Z. Xu, and Q. Zhao, "A fountain-coding based cooperative jamming strategy for secure service migration in edge computing," *Wireless Networks*, pp. 1–14, 2021.

[158] A. Braeken, P. Porambage, A. Puvaneswaran, and M. Liyanage, "Essmar: Edge supportive secure mobile augmented reality architecture for healthcare," in *2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech)*. IEEE, 2020, pp. 1–7.

[159] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[160] A. K. Yadav, M. Misra, P. K. Pandey, A. Braeken, and M. Liyange, "An improved and provably secure symmetric-key based 5g-aka protocol," *Computer Networks*, vol. 218, p. 109400, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128622004340

[161] A. K. Yadav, M. Misra, P. K. Pandey, and M. Liyanage, "An eap-based mutual authentication protocol for wlan connected iot devices," *IEEE Transactions on Industrial Informatics*, pp. 1–12, 2022.

[162] A. Mahmood, M. I. Ashraf, M. Gidlund, J. Torsner, and J. Sachs, "Time synchronization in 5g wireless edge: Requirements and solutions for critical-mtc," *IEEE Communications Magazine*, vol. 57, no. 12, pp. 45–51, 2019.

[163] S. Ravi, S. Trehan, M. Jain, and H. M. Kittur, "High performance clock path elements for clock skew reduction," in *2019 2nd international conference on intelligent computing, instrumentation and control technologies (ICICICT)*, vol. 1. IEEE, 2019, pp. 1663–1670.

[164] S. Son, J. Lee, Y. Park, Y. Park, and A. K. Das, "Design of blockchain-based lightweight v2i handover authentication protocol for vanet," *IEEE Transactions on Network Science and Engineering*, 2022.

[165] T. Aura, P. Nikander, and J. Leiwo, "Dos-resistant authentication with client puzzles," in *International workshop on security protocols*. Springer, 2000, pp. 170–177.

[166] L. D. Tsobdjou, S. Pierre, and A. Quintero, "A new mutual authentication and key agreement protocol for mobile client—server environment," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1275–1286, 2021.

[167] J. Zhang, J. Cui, H. Zhong, Z. Chen, and L. Liu, "Pa-crt: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 722–735, 2019.

[168] C. J. F. Cremers, *Scyther: Semantics and verification of security protocols*. Eindhoven university of Technology Eindhoven, Netherlands, 2006.

[169] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. H. Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani *et al.*, "The avispa tool for the automated validation of internet security protocols and applications," in *International conference on computer aided verification*. Springer, 2005, pp. 281–285.

[170] Y. Zhang, X. Chen, J. Li, and H. Li, "Generic construction for secure and efficient handoff authentication schemes in eap-based wireless networks," *Computer Networks*, vol. 75, pp. 192–211, 2014.

[171] L. Gong, R. M. Needham, and R. Yahalom, "Reasoning about belief in cryptographic protocols." in *IEEE Symposium on Security and Privacy*. Citeseer, 1990, pp. 234–248.

[172] M. Abdalla, P.-A. Fouque, and D. Pointcheval, "Password-based Authenticated Key Exchange in the Three-party Setting," in *International Workshop on Public Key Cryptography*. Springer, 2005, pp. 65–84.

[173] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *International conference on the theory and applications of cryptographic techniques*. Springer, 2000, pp. 139–155.

[174] P. K. Roy, P. Sahu, and A. Bhattacharya, "Fasthand: A fast handover authentication protocol for densely deployed small-cell networks," *Journal of Network and Computer Applications*, p. 103435, 2022.

[175] S. Gupta, B. L. Parne, N. S. Chaudhari, and S. Saxena, "Seai: Secrecy and efficiency aware inter-gnb handover authentication and key agreement protocol in 5g communication network," *Wireless Personal Communications*, vol. 122, no. 4, pp. 2925–2962, 2022.

[176] M. Ramadan, F. Li, C. Xu, A. Mohamed, H. Abdalla, and A. A. Ali, "User-to-user mutual authentication and key agreement scheme for lte cellular system." *Int. J. Netw. Secur.*, vol. 18, no. 4, pp. 769–781, 2016.

[177] J. Choi and S. Jung, "A handover authentication using credentials based on chameleon hashing," *IEEE communications letters*, vol. 14, no. 1, pp. 54–56, 2009.

[178] Z. Zhou, H. Zhang, and Z. Sun, "An improved privacy-aware handoff authentication protocol for vanets," *Wireless personal communications*, vol. 97, no. 3, pp. 3601–3618, 2017.

[179] G. Yang, Q. Huang, D. S. Wong, and X. Deng, "Universal authentication protocols for anonymous wireless communications," *IEEE Transactions on Wireless Communications*, vol. 9, no. 1, pp. 168–174, 2010.

[180] D. He, C. Chen, S. Chan, and J. Bu, "Secure and efficient handover authentication based on bilinear pairing functions," *IEEE Transactions on Wireless Communications*, vol. 11, no. 1, pp. 48–53, 2011.

[181] D. He, J. Bu, S. Chan, C. Chen, and M. Yin, "Privacy-preserving universal authentication protocol for wireless communications," *IEEE transactions on wireless communications*, vol. 10, no. 2, pp. 431–436, 2010.

[182] D. He, D. Wang, Q. Xie, and K. Chen, "Anonymous handover authentication protocol for mobile wireless networks with conditional privacy preservation," *Science China Information Sciences*, vol. 60, no. 5, pp. 1–17, 2017.

[183] Z. Xu, X. Li, J. Xu, W. Liang, and K.-K. R. Choo, "A secure and computationally efficient authentication and key agreement scheme for internet of vehicles," *Computers & Electrical Engineering*, vol. 95, p. 107409, 2021.

[184] P. Arthurs, L. Gillam, P. Krause, N. Wang, K. Halder, and A. Mouzakitis, "A taxonomy and survey of edge cloud computing for intelligent transportation systems and connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[185] X. Yu, M. Guan, M. Liao, and X. Fan, "Pre-migration of vehicle to network services based on priority in mobile edge computing," *IEEE Access*, vol. 7, pp. 3722–3730, 2018.

[186] Q. Yuan, J. Li, H. Zhou, T. Lin, G. Luo, and X. Shen, "A joint service migration and mobility optimization approach for vehicular edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 9041–9052, 2020.

[187] S. Ge, M. Cheng, X. He, and X. Zhou, "A two-stage service migration algorithm in parked vehicle edge computing for internet of things," *Sensors*, vol. 20, no. 10, p. 2786, 2020.

[188] J. Xu, X. Ma, A. Zhou, Q. Duan, and S. Wang, "Path selection for seamless service migration in vehicular edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 9040–9049, 2020.

[189] S. Moon and Y. Lim, "Task migration with partitioning for load balancing in collaborative edge computing," *Applied Sciences*, vol. 12, no. 3, p. 1168, 2022.

[190] Z. Liu and X. Xu, "Latency-aware service migration with decision theory for internet of vehicles in mobile edge computing," *Wireless Networks*, pp. 1–13, 2022.

[191] A. Lakhan, M. Ahmad, M. Bilal, A. Jolfaei, and R. M. Mehmood, "Mobility aware blockchain enabled offloading and scheduling in vehicular fog cloud computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4212–4223, 2021.

[192] X. Xiao, Y. Ma, Y. Xia, M. Zhou, X. Luo, X. Wang, X. Fu, W. Wei, and N. Jiang, "Novel workload-aware approach to mobile user reallocation in crowded mobile edge computing environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 8846–8856, 2022.

[193] W. Wang, S. Ge, and X. Zhou, "Location-privacy-aware service migration in mobile edge computing," in *2020 IEEE wireless communications and networking conference (WCNC)*. IEEE, 2020, pp. 1–6.

[194] Y. Zhong, X. Ge, H. H. Yang, T. Han, and Q. Li, "Traffic matching in 5g ultra-dense networks," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 100–105, 2018.

[195] X. Ge, S. Tu, G. Mao, C.-X. Wang, and T. Han, "5g ultra-dense cellular networks," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 72–79, 2016.

[196] H. Wang, Y. Huang, A. Khajepour, Y. Zhang, Y. Rasekhipour, and D. Cao, "Crash mitigation in motion planning for autonomous vehicles," *IEEE transactions on intelligent transportation systems*, vol. 20, no. 9, pp. 3313–3323, 2019.

[197] A. Naser, M. Gavahi, C. Wu, V. T. Hoang, Z. Wang, and X. Yuan, "An empirical study of cryptographic libraries for mpi communications," in *2019 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2019, pp. 1–11.

[198] M. N. Khan, A. Rao, and S. Camtepe, "Lightweight cryptographic protocols for iot-constrained devices: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4132–4156, 2020.

[199] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge computing based on markov decision process," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1272–1288, 2019.

[200] I. A. M. Balapuwaduge, L. Jiao, V. Pla, and F. Li, "Channel assembling with priority-based queues in cognitive radio networks: Strategies and performance evaluation," *IEEE Transactions on Wireless Communications*, vol. 13, no. 2, pp. 630–645, 2014.

[201] M. Ochse, "T-Pot - The All In One Multi Honeypot Platform," 2022. [Online]. Available: https://github.com/telekom-security/tpotce.git

[202] Y. Khettab, M. Bagaa, D. L. C. Dutra, T. Taleb, and N. Toumi, "Virtual Security as a Service for 5G Verticals," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2018, pp. 1–6.

[203] A. Boudi, I. Farris, M. Bagaa, and T. Taleb, "Assessing lightweight virtualization for security-as-a-service at the network edge," *IEICE Transactions on Communications*, vol. 102, no. 5, pp. 970–977, 2019.

[204] A. Sforzin, F. G. Mármol, M. Conti, and J.-M. Bohli, "Rpids: Raspberry pi ids—a fruitful intrusion detection system for iot," in *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*. IEEE, 2016, pp. 440–448.

[205] S. Tripathi and R. Kumar, "Raspberry pi as an intrusion detection system, a honeypot and a packet analyzer," in *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*. IEEE, 2018, pp. 80–85.

[206] B. I. Ismail, E. M. Goortani, M. B. Ab Karim, W. M. Tat, S. Setapa, J. Y. Luke, and O. H. Hoe, "Evaluation of docker as edge computing platform," in *2015 IEEE Conference on Open Systems (ICOS)*. IEEE, 2015, pp. 130–135.

[207] K. Nam and K. Kim, "A study on sdn security enhancement using open source ids/ips suricata," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2018, pp. 1124–1126.

[208] H. M. Elshafie, T. M. Mahmoud, and A. A. Ali, "Improving the performance of the snort intrusion detection using clonal selection," in *2019*

International Conference on Innovative Trends in Computer Engineering (ITCE). IEEE, 2019, pp. 104–110.

[209] A. D. Pinto. (2019, Jan) Tricotools. [Online]. Available: https://github.com/NozomiNetworks/tricotools

[210] T. Miao and H. Chen, "Flexcore: Dynamic virtual machine scheduling using vcpu ballooning," *Tsinghua Science and Technology*, vol. 20, no. 1, pp. 7–16, 2015.

[211] J. Park, D. Lee, B. Kim, J. Huh, and S. Maeng, "Locality-aware dynamic vm reconfiguration on mapreduce clouds," in *Proceedings of the 21st international symposium on High-Performance Parallel and Distributed Computing*. ACM, 2012, pp. 27–36.

[212] A. Lingayat, R. R. Badre, and A. K. Gupta, "Performance evaluation for deploying docker containers on baremetal and virtual machine," in *2018 3rd International Conference on Communication and Electronics Systems (ICCES)*. IEEE, 2018, pp. 1019–1023.

[213] R. Kumar and B. Thangaraju, "Performance analysis between runc and kata container runtime," in *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*. IEEE, 2020, pp. 1–4.

[214] C. Fan and L. Li, "Service migration in mobile edge computing based on reinforcement learning," in *Journal of Physics: Conference Series*, vol. 1584, no. 1. IOP Publishing, 2020, p. 012058.

[215] D. Zhao, T. Yang, Y. Jin, and Y. Xu, "A service migration strategy based on multiple attribute decision in mobile edge computing," in *2017 IEEE 17th international conference on communication technology (ICCT)*. IEEE, 2017, pp. 986–990.

[216] A. Algarni, M. Ahmad, A. Attaallah, A. Agrawal, R. Kumar, and R. A. Khan, "A fuzzy multi-objective covering-based security quantification

model for mitigating risk of web based medical image processing system," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 1, 2020.

[217] Z. Zeng, J. Zhu, H. Qiu, and T. Zhou, "Sm-rc: A new security measurement method for inter-domain routing system," *IEEE Access*, vol. 7, pp. 108 189–108 199, 2019.

APPENDIX

ONE

SUPPLEMENTARY APPENDICES FOR CHAPTER 5: MEC SERVICE MIGRATION AUTHENTICATION PROTOCOL

The aim of the appendix is to grant the reviewers a holistic perspective of the undergone implementations of this protocol, complete validation scripts and results, and any other content that was not able to facilitate within the thesis. This document contains the following.

1. Appendix A.1: Details of the Message Identification Headers
2. Appendix A.2: SPDL Scripts employed for Scyther-based Validations
3. Appendix A.3: HLPSL Scripts employed for AVISPA-based Validations and their Results
4. Appendix A.4: DoS Puzzle
5. Appendix A.5: Protocol Details of the Legacy Protocol
6. Appendix A.6: MatLab Code for the Simulation in Fig. 5.16
7. Appendix A.7: Details of the MEC Prototype Development Environment

Table A.1: The MIH Specifications of the Proposed SP

Source Entity		Message Specification		Payload Specification	
TTP	Trusted Third Party Server	MR	Migration Registration	REQ	Request
S	Source gNB	MA	Migration Authentication	REP	Reply
R	Roaming gNB	MS	Migration Session	INIT	Initialization
MVR	MES Verification Registry	MES	Mobile Edge Service Verification	HELLO	First Contact
CA	Certificate Authority	RES	MES Resource Verification	VER	Verification
OSS	Operations Support System	TD	TTP Discovery		
		SV	Source Verification		

A.1 Details of the Message Identification Headers

The Proposed Standard Protocol (SP) in the paper includes Message Identification Headers (MIHs) throughout the protocol description for the sake of identifying/ distinguishing the messages in the protocol operation. Each MIH contains three specifications. The format takes the form: $<$*Source/Message Originating Entity*$>$_$<$*Message Specification*$>$_$<$*Payload Specification*$>$ (e.g. OSS_TD_REQ). Table A.1 describes the various MIH entries used in the proposed protocol and their meaning.

A.2 SPDL Scripts employed for Scyther-based Validations

Scyther validation tool was utilized for verifying the proposed protocol formally against known attacks as a model-based verification approach. As indicated in the paper, Scyther employs a language called SPDL for specifying the protocols. We have successfully validated our protocol for the segregated phases

(i.e. Phases A through F). In all instances, we executed the tool with the specified settings of 100 runs, as indicated in Fig. A.1. And the following subsections are specifying the SPDL scripts used to perform the validation along with their corresponding results.

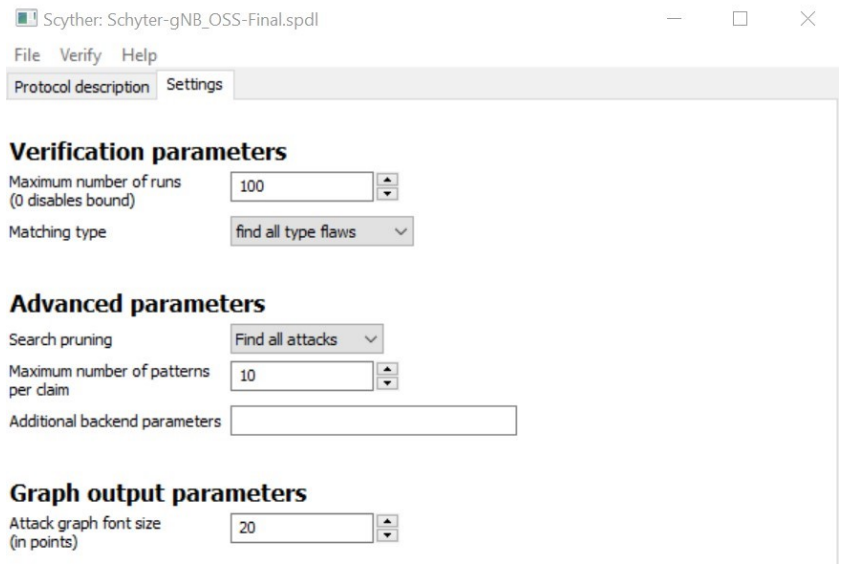


Figure A.1: Specifications of the Scyther Tools' Settings for the Conducted Validation Scenarios

A.2.1 Part A: gNB_S and OSS Communication

Listing A.1 and Fig. A.2 are specifying and depict the SPDL specification and the corresponding verification result of Part A respectively.

```
# MEC Service Migration gNB_S-OSS Communication
#This version is specified in accordance with the implementation followed
    in the protocol

#usertype SessionKey;
const Fresh: Function;
const Multiply: Function;
usertype Timestamp;
usertype String;
```

```
#Hash Function Declaration
hashfunction H;

protocol gNBs-OSS(gNBs,OSS)
{
  #Message Headers/ IDs
  const Hello: String;
  const OSS-TD-REP: String;
  const S-TD-REQ: String;
  const OSS-TD-REP1: String;

  #String Variables
  const k-dos: String;
  const M: String;
  const X: String;
  const Certificate-TTP: String;    #TTP public key
  const SOCK-TTP: String;     #TTP Socket or API address

    role gNBs
    {
  #Nonces
    var Noss: Nonce;
    fresh Ns: Nonce;
    fresh a1: Nonce;
    var b1: Nonce;
  #Timestamps
    fresh T1: Timestamp;
    fresh T2: Timestamp;
    fresh T3: Timestamp;
    fresh T4: Timestamp;
  #Other Macro Computations
    macro Nv1 = H(Noss,T3);       #Nonce Verifier 1
    macro Nv2 = H(Ns,T4);       #Nonce Verifier 2
    macro PA1 = Multiply(a1,M);
    macro PA2 = Multiply(b1,M);
    macro SyKA = Multiply(a1,b1,M);
  #Signature Hashes
    macro SigH1 = H(OSS,T1);
    macro SigH2 = H(gNBs,T2);
```

```
  #HMAC Computations
    macro hmac1 = H(gNBs,OSS,T1);
    macro hmac2 = H(k-dos,Noss,gNBs,T2);

#Message Specifications
send_1(gNBs,OSS, Hello, {gNBs,PA1,T1}pk(OSS), {SigH1}sk(gNBs), hmac1);
recv_2(OSS,gNBs, {OSS-TD-REP, k-dos, Noss, PA2,T2}pk(gNBs), {SigH2}sk(OSS)
    , hmac2);
match(hmac2,H(k-dos,Noss,gNBs,PA2,T2));
send_3(gNBs,OSS, {S-TD-REQ, Ns, X, Nv1, T3}pk(OSS));
recv_4(OSS,gNBs, {OSS-TD-REP1, {SOCK-TTP, Certificate-TTP}SyKA ,Nv2,T4}pk(
    gNBs));
match(Nv2, H(Ns,T4));

#Claims
claim_gNBs1(gNBs,Nisynch);
claim_gNBs2(gNBs,Niagree);
claim_gNBs3(gNBs,Empty,(Fresh,T2));
claim_gNBs4(gNBs,Secret,Noss);
    }



    role OSS
    {
  #Nonces
    fresh Noss: Nonce;
    var Ns: Nonce;
    var a1: Nonce;
    fresh b1: Nonce;
  #Timestamps
    fresh T1: Timestamp;
    fresh T2: Timestamp;
    fresh T3: Timestamp;
    fresh T4: Timestamp;
  #HMACs
    var hmac1: String;
  #Macro Computations
    macro hmac2 = H(k-dos,Noss,gNBs,T2);
```

```
    macro Nv1 = H(Noss,T3);          #Nonce Verifier 1
    macro Nv2 = H(Ns,T4);          #Nonce Verifier
    macro PA1 = Multiply(a1,M);
    macro PA2 = Multiply(b1,M);
    macro SyKA = Multiply(a1,b1,M);
  #Signature Hashes
    macro SigH1 = H(OSS,T1);
    macro SigH2 = H(gNBs,T2);

#Message Specifications
recv_1(gNBs,OSS, Hello, {gNBs,PA1,T1}pk(OSS), {SigH1}sk(gNBs), hmac1);
match(hmac1,H(gNBs,OSS,PA1,T1));
send_2(OSS,gNBs, {OSS-TD-REP, k-dos, Noss,PA2,T2}pk(gNBs), {SigH2}sk(OSS),
     hmac2);
recv_3(gNBs,OSS, {S-TD-REQ, Ns, X, Nv1, T3}pk(OSS));
match(Nv1, H(Noss,T3));
send_4(OSS,gNBs, {OSS-TD-REP1, {SOCK-TTP, Certificate-TTP}SyKA, Nv2,T4}pk(
    gNBs));

#Claims
claim_OSS1(OSS,Nisynch);
claim_OSS2(OSS,Niagree);
claim_OSS3(OSS,Empty,(Fresh,T1));
claim_OSS1(OSS, Secret,Ns);
    }
};
```

Listing A.1: Part-A Scyther Script

A.2.2 Part B: gNB_S and TTP Communication

Listing A.2 and Fig. A.3 are specifying and depict the SPDL specification and the corresponding verification result of Protocol section B respectively.

```
# MEC Service Migration B : gNBs-TTP Communication Version Final
#usertype SessionKey;
const Fresh: Function;
const Multiply: Function;
usertype Timestamp;
```

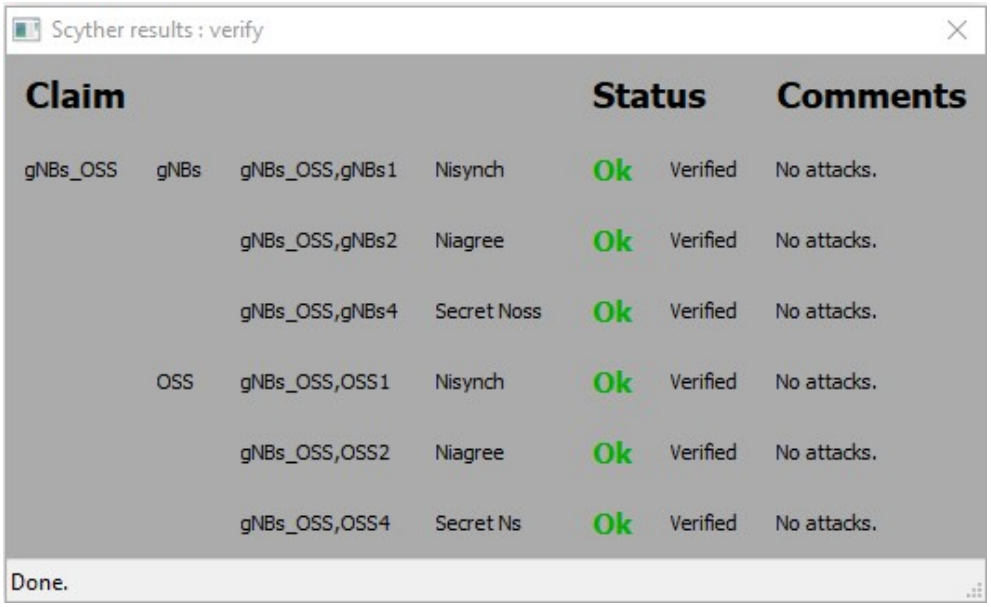


Figure A.2: Scyther Validation Results for the Part A of the SP

```
usertype String;

#Hash Function Declaration
hashfunction H;

protocol gNBs-TTP(gNBs,TTP)
{
  #Message Headers/ IDs
  const Hello: String;
  const TTP-MR-REP: String;
  const S-MR-REQ: String;
  const TTP-MR-REP1: String;
  #String Variables/ Constants
  const k-dos: String;
  const M: String;        #ECC Public Point
  const X: String;
  const MP-ID-ARR: String;      #Migration Process ID Array
  const TTP-M-SOCK: String;     #TTP Migration Socket or API address

    role gNBs
    {
```

```
  #Nonces
    var Nttp: Nonce;
    fresh Ns: Nonce;
    fresh a2: Nonce;
    var b2: Nonce;
    var r1dash: Nonce;
  #Timestamps
    fresh T1: Timestamp;
    fresh T2: Timestamp;
    fresh T3: Timestamp;
    fresh T4: Timestamp;
  #Other Macro Computations
    macro Nv1 = H(Nttp,T3);          #Nonce Verifier 1
    macro Nv2 = H(Ns,T4);          #Nonce Verifier 2
    macro PB1 = Multiply(a2,M);
    macro PB2 = Multiply(b2,M);
    macro SyKB = Multiply(a2,b2,M);
  #Signature Hashes
    macro SigH1 = H(TTP,T1);
    macro SigH2 = H(gNBs,T2);
  #HMAC Computations
    macro hmac1 = H(gNBs,PB1,TTP,T1);
    macro hmac2 = H(k-dos,Nttp,gNBs,PB2,T2);
#Message Specifications
send_1(gNBs,TTP, Hello,{gNBs, PB1, T1}pk(TTP), {SigH1}sk(gNBs),hmac1);
recv_2(TTP,gNBs, {TTP-MR-REP, k-dos, Nttp, PB2, T2}pk(gNBs), {SigH2}sk(TTP
    ), hmac2);
match(hmac2,H(k-dos,Nttp,gNBs,PB2,T2));
send_3(gNBs,TTP, {S-MR-REQ, Ns, X, Nv1, T3}pk(TTP));
recv_4(TTP,gNBs, {TTP-MR-REP1, {TTP-M-SOCK, MP-ID-ARR, r1dash}SyKB, Nv2,
    T4}pk(gNBs));
match(Nv2, H(Ns,T4));
#Claims
claim_gNBs1(gNBs,Nisynch);
claim_gNBs2(gNBs,Niagree);
claim_gNBs3(gNBs,Empty,(Fresh,T2));
claim_gNBs4(gNBs,Secret,Nttp);
    }
```

```
    role TTP
    {
  #Nonces
    fresh Nttp: Nonce;
    var Ns: Nonce;
    var a2: Nonce;
    fresh b2: Nonce;
    fresh r1dash: Nonce;
  #Timestamps
    fresh T1: Timestamp;
    fresh T2: Timestamp;
    fresh T3: Timestamp;
    fresh T4: Timestamp;
  #Other Macro Computations
    macro Nv1 = H(Nttp,T3);          #Nonce Verifier 1
    macro Nv2 = H(Ns,T4);          #Nonce Verifier 2
    macro PB1 = Multiply(a2,M);
    macro PB2 = Multiply(b2,M);
    macro SyKB = Multiply(a2,b2,M);
  #Signature Hashes
    macro SigH1 = H(TTP,T1);
    macro SigH2 = H(gNBs,T2);
  #HMAC Computations
    macro hmac1 = H(gNBs,PB1,TTP,T1);
    macro hmac2 = H(k-dos,Nttp,gNBs,PB2,T2);
#Message Specifications
recv_1(gNBs,TTP, Hello,{gNBs, PB1, T1}pk(TTP), {SigH1}sk(gNBs),hmac1);
match(hmac1,H(gNBs,PB1,TTP,T1));
send_2(TTP,gNBs, {TTP-MR-REP, k-dos, Nttp, PB2, T2}pk(gNBs), {SigH2}sk(TTP
    ), hmac2);
recv_3(gNBs,TTP, {S-MR-REQ, Ns, X, Nv1, T3}pk(TTP));
match(Nv1, H(Nttp,T3));
send_4(TTP,gNBs, {TTP-MR-REP1, {TTP-M-SOCK, MP-ID-ARR, r1dash}SyKB, Nv2,
    T4}pk(gNBs));
#Claims
claim_TTP1(TTP,Nisynch);
claim_TTP2(TTP,Niagree);
claim_TTP3(TTP,Empty,(Fresh,T1));
claim_TTP4(TTP, Secret,Ns);
```

```
    }
};
```

Listing A.2: Part-B Scyther Script

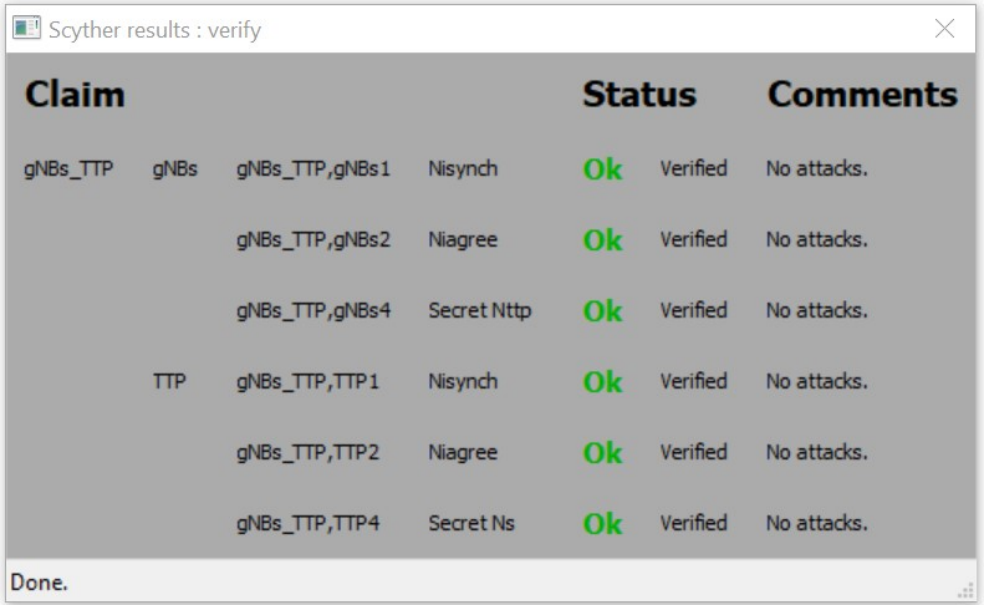


Figure A.3: Scyther Validation Results for the Part B of the SP

A.2.3 Part C&D: gNB_S, gNB_R and TTP Communication for Service Migration Registration

Listing A.3 and Fig. A.4 are specifying and depict the SPDL specification and the corresponding verification result of the Protocol sections C & D respectively.

```
# MEC Service Migration C & D : gNBs-gNBr-TTP Communication Version Final
#usertype SessionKey;
const Fresh: Function;
const Multiply: Function;
usertype Timestamp;
usertype String;

#Hash Function Declaration
```

```
hashfunction H;

protocol gNBs-gNBr-TTP(gNBs,gNBr,TTP)
{
  #Message Headers/ IDs
  const S-MA-REQ: String;
  const R-SV-REQ: String;
  const TTP-SV-REP: String;
  const TTP-SV-VER: String;
  const R-MA-REP: String;
  #String Variables/ Constants
  const k-dos: String;
  const M: String;
  const Certificate-TTP: String;    #Certificate of the TTP
  const MP-ID: String;        #Migration Process ID
  const TTP-M-SOCK: String;     #TTP Migration Socket or API address

    role gNBs
    {
  #Nonces and Variables
    fresh Ns: Nonce;
    var r2dash: Nonce;
    var Nr: Nonce;
    var RAND: Nonce;
    fresh a3: Nonce;
    var b3: Nonce;
    var a4: Nonce;
    var b4: Nonce;
    var c: Nonce;
  #Timestamps
    fresh T1: Timestamp;
    fresh T4: Timestamp;
    fresh T5: Timestamp;
  #Other Macro Computations
    macro CODE-M = H(gNBs,gNBr,MP-ID,RAND); #CODE-M for verification
    macro Nv2 = H(Ns,T5);         #Nonce Verifier 2
    macro PC1 = Multiply(a3,M);
    macro PC2 = Multiply(b3,M);
    macro PC3 = Multiply(c,M);
```

```
    macro SyKC1 = Multiply(a3,b3,M);
    macro SyKC2 = Multiply(a3,c,M);
  #HMAC Computations
    macro hmac1 = H(gNBs,TTP,MP-ID,TTP-M-SOCK, Certificate-TTP, Ns, PC1,
    gNBr, T1);
#Message Specifications
send_1(gNBs,gNBr, S-MA-REQ,{gNBs,TTP,MP-ID,TTP-M-SOCK,Certificate-TTP, Ns,
     PC1, T1}pk(gNBr), {H(gNBr,T1)}sk(gNBs),hmac1);
recv_4(TTP,gNBs, TTP-SV-VER,{TTP,MP-ID,PC2, {CODE-M,r2dash}SyKC1, T4}pk(
    gNBs));
recv_5(gNBr,gNBs, R-MA-REP,{gNBr,MP-ID,PC3, {CODE-M,Nr}SyKC2, Nv2,T5}pk(
    gNBs),{H(gNBs,T5)}sk(gNBr));
match(Nv2, H(Ns,T5));
#Claims
claim_gNBs1(gNBs,Nisynch);
claim_gNBs2(gNBs,Niagree);
claim_gNBs3(gNBs,Empty,(Fresh,T4));
claim_gNBs4(gNBs,Empty,(Fresh,T5));
claim_gNBs5(gNBs,Secret,Nr);
}

role gNBr
{
  #Nonces and Variables
    var Ns: Nonce;
    var N: Nonce;
    var r1: Nonce;
    var r2: Nonce;
    fresh Nr: Nonce;
    var RAND:Nonce;
    var a3: Nonce;
    fresh a4: Nonce;
    var b4: Nonce;
    fresh c: Nonce;
  #Timestamps
    fresh T1: Timestamp;
    fresh T2: Timestamp;
    fresh T3: Timestamp;
    fresh T4: Timestamp;
```

```
    fresh T5: Timestamp;
  #Other Macro Computations
    macro CODE-M = H(gNBs,gNBr,MP-ID,RAND); #CODE-M for verification
    macro Nv1 = H(Nr,T3);          #Nonce Verifier 1
    macro Nv2 = H(Ns,T5);          #Nonce Verifier 2
    macro PC1 = Multiply(a3,M);
    macro PC3 = Multiply(c,M);
    macro PD1 = Multiply(a4,M);
    macro PD2 = Multiply(b4,M);
    macro SyKD = Multiply(a4,b4,M);
    macro SyKC2 = Multiply(a3,c,M);
  #HMAC Computations
    macro hmac1 = H(gNBs,TTP,MP-ID,TTP-M-SOCK, Certificate-TTP, Ns, PC1,
    gNBr, T1);
    macro hmac2 = H(gNBr,TTP,MP-ID,Nr,PC1,PD1,T2);
#Message Specifications
recv_1(gNBs,gNBr, S-MA-REQ,{gNBs,TTP,MP-ID,TTP-M-SOCK,Certificate-TTP, Ns,
     PC1, T1}pk(gNBr), {H(gNBr,T1)}sk(gNBs),hmac1);
match(hmac1,H(gNBs,TTP,MP-ID,TTP-M-SOCK, Certificate-TTP, Ns, PC1, gNBr,
    T1));
send_2(gNBr,TTP, R-SV-REQ,{gNBr,TTP,MP-ID,Nr,PC1,PD1,T2}pk(TTP),{H(TTP,T2)
    }sk(gNBr),hmac2);
recv_3(TTP,gNBr, TTP-SV-REP,{MP-ID,PD2, {CODE-M,r1,r2,N}SyKD, Nv1,T3}pk(
    gNBr),{H(gNBs,T3)}sk(TTP));
match(Nv1, H(Nr,T3));
send_5(gNBr,gNBs,R-MA-REP,{gNBr,MP-ID,PC3, {CODE-M,Nr}SyKC2, Nv2,T5}pk(
    gNBs),{H(gNBs,T5)}sk(gNBr));
#Claims
claim_gNBr1(gNBr,Nisynch);
claim_gNBr2(gNBr,Niagree);
claim_gNBr3(gNBr,Empty,(Fresh,T1));
claim_gNBr4(gNBr,Empty,(Fresh,T3));
claim_gNBr5(gNBr,Secret,Ns);
}
role TTP
{
  #Nonces
    var Nr: Nonce;
    fresh r1: Nonce;
```

```
    fresh r2: Nonce;
    fresh r2dash: Nonce;
    fresh N: Nonce;
    fresh RAND: Nonce;
    fresh b3: Nonce;
    var a4: Nonce;
    fresh b4: Nonce;
    var a3: Nonce;
  #Timestamps
    fresh T2: Timestamp;
    fresh T3: Timestamp;
    fresh T4: Timestamp;
  #Other Macro Computations
    macro Nv1 = H(Nr,T3);          #Nonce Verifier 1
    macro CODE-M = H(gNBs,gNBr,MP-ID,RAND); #CODE-M for verification
    macro PC1 = Multiply(a3,M);
    macro PC2 = Multiply(b3,M);
    macro PD1 = Multiply(a4,M);
    macro PD2 = Multiply(b4,M);
    macro SyKD = Multiply(a4,b4,M);
    macro SyKC1 = Multiply(a3,b3,M);
  #HMAC Computations
    macro hmac2 = H(gNBr,TTP,MP-ID,Nr,PC1,PD1,T2);
#Message Specifications
recv_2(gNBr,TTP, R-SV-REQ,{gNBr,TTP,MP-ID,Nr,PC1,PD1,T2}pk(TTP), {H(TTP,T2
    )}sk(gNBr),hmac2);
match(hmac2,H(gNBr,TTP,MP-ID,Nr,PC1,PD1,T2));
send_3(TTP,gNBr, TTP-SV-REP,{MP-ID,PD2, {CODE-M,r1,r2,N}SyKD, Nv1,T3}pk(
    gNBr),{H(gNBs,T3)}sk(TTP));
send_4(TTP,gNBs, TTP-SV-VER,{TTP,MP-ID,PC2, {CODE-M,r2dash}SyKC1, T4}pk(
    gNBs));
#Claims
claim_TTP1(TTP,Nisynch);
claim_TTP2(TTP,Niagree);
claim_TTP3(TTP,Empty,(Fresh,T2));
claim_TTP4(TTP, Secret,Nr);
}
};
```

Listing A.3: Parts - C and D Scyther Script

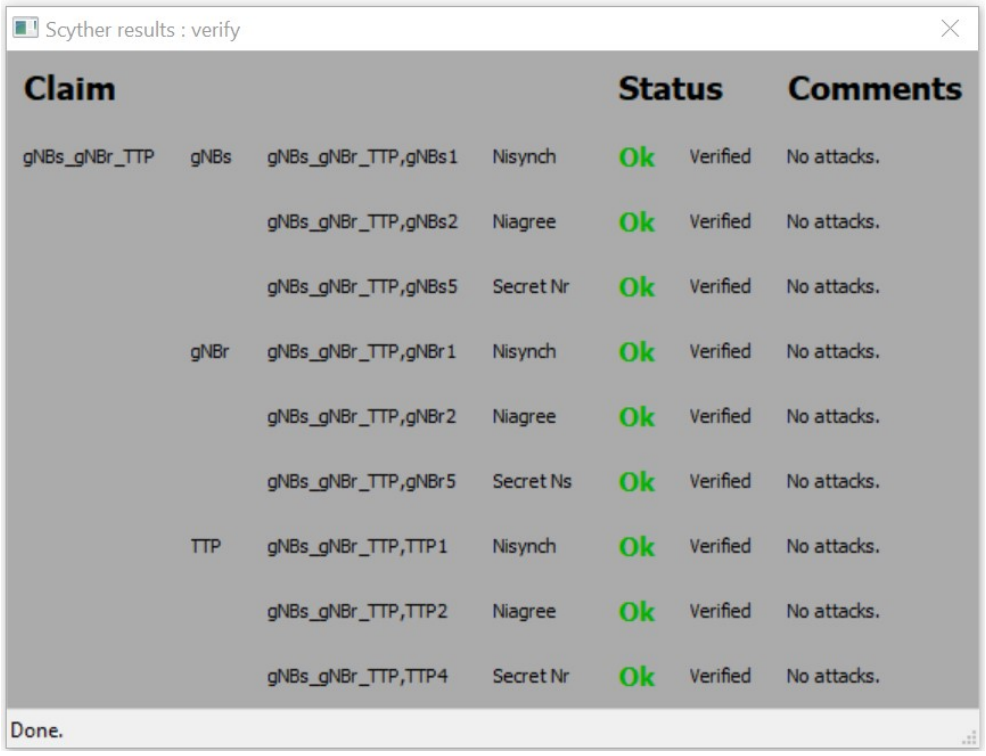


Figure A.4: Scyther Validation Results for the Part C and Part D of the SP

A.2.4 Part E&F: gNB_S, gNB_R and MVR Communication for MES Verification

Listing A.4 and Fig. A.5 are specifying and depict the SPDL specification and the corresponding verification result of the Protocol sections E & F respectively.

```
# MEC Service Migration E & F : gNBs-gNBr-MVR Communication Version Final
#usertype SessionKey;
const Fresh: Function;
const Multiply: Function;
usertype Timestamp;
usertype String;

#Hash Function Declaration
hashfunction H;

protocol gNBs-gNBr-MVR(gNBs,gNBr,MVR)
```

```
{
  #Message Headers/ IDs
  const S-MES-REQ: String;
  const R-MES-REQ: String;
  const MVR-MES-REP: String;
  const R-MES-REP: String;
  const S-MES-VER: String;
  #String Variables/ Constants
  const MP-ID: String;        #Migration Process ID
  const M: String;
  const MES-ID: String;
  const MES-STATE: String;
  const MES-DATA: String;
  const MES-REQ: String;
  const MES-VER: String;
  const MES-RES: String;

    role gNBs
    {
  #Nonces and Variables
    var RAND: Nonce;
    fresh a5: Nonce;
    var a6: Nonce;
    var a7: Nonce;
  #Timestamps
    fresh T1: Timestamp;
    fresh T4: Timestamp;
    fresh T5: Timestamp;
  #Other Macro Computations
    macro CODE-MES = H(MES-ID,MES-STATE,MVR,gNBs,gNBr,RAND);  #CODE-MES
    for verification
    macro PE1 = Multiply(a5,M);
    macro PF1 = Multiply(a6,M);
    macro PF2 = Multiply(a7,M);
    macro SyKE1 = Multiply(a5,a6,M);
    macro SyKE2 = Multiply(a5,a7,M);
  #HMAC Computations
    macro hmac1 = H(MP-ID,MES-ID,MES-STATE,MES-DATA,MES-REQ,MVR,PE1,T1);
    #macro hmac4 = H(gNBr,MES-RES,CODE-MES,T4);
```

```
    #macro hmac5 = H(gNBs,MES-ID,CODE-MES,T5);
#Message Specifications
send_1(gNBs,gNBr, S-MES-REQ,{MP-ID,MES-ID,MES-STATE,MES-DATA,MES-REQ,MVR,
    PE1,T1}pk(gNBr),hmac1);
recv_4(gNBr,gNBs, R-MES-REP,{gNBr,PF1,PF2, {MES-RES,CODE-MES}SyKE1, T4}pk(
    gNBs));
send_5(gNBs,MVR, S-MES-VER,{gNBs,MES-ID, {CODE-MES}SyKE2, T5}pk(MVR));
#Claims
claim_gNBs1(gNBs,Nisynch);
claim_gNBs2(gNBs,Niagree);
claim_gNBs3(gNBs,Empty,(Fresh,T4));
  }

  role gNBr
  {
  #Nonces and Variables
    fresh Nr: Nonce;
    var RAND:Nonce;
    var a5: Nonce;
    fresh a6: Nonce;
    var a7: Nonce;
  #Timestamps
    fresh T1: Timestamp;
    fresh T2: Timestamp;
    fresh T3: Timestamp;
    fresh T4: Timestamp;
  #Other Macro Computations
    macro CODE-MES = H(MES-ID,MES-STATE,MVR,gNBs,gNBr,RAND);  #CODE-MES
    for verification
    macro Nv1 = H(Nr,T3);          #Nonce Verifier 1
    macro PE1 = Multiply(a5,M);
    macro PF1 = Multiply(a6,M);
    macro PF2 = Multiply(a7,M);
    macro SyKF = Multiply(a6,a7,M);
    macro SyKE1 = Multiply(a5,a6,M);
  #HMAC Computations
    macro hmac1 = H(MP-ID,MES-ID,MES-STATE,MES-DATA,MES-REQ,MVR,PE1,T1);
    macro hmac2 = H(MES-ID,MES-STATE,gNBs,gNBr,Nr,PE1,PF1,MVR,T2);
#Message Specifications
```

```
recv_1(gNBs,gNBr, S-MES-REQ,{MP-ID,MES-ID,MES-STATE,MES-DATA,MES-REQ,MVR,
    PE1,T1}pk(gNBr),hmac1);
match(hmac1,H(MP-ID,MES-ID,MES-STATE,MES-DATA,MES-REQ,MVR,PE1,T1));
send_2(gNBr,MVR, R-MES-REQ,{MES-ID,MES-STATE,gNBs,gNBr,Nr,PE1,PF1,T2}pk(
    MVR),{H(MVR,T2)}sk(gNBr),hmac2);
recv_3(MVR,gNBr, MVR-MES-REP,{MVR,PF2, {MES-VER,CODE-MES}SyKF, Nv1,T3}pk(
    gNBr),{H(gNBr,T3)}sk(MVR));
match(Nv1, H(Nr,T3));
send_4(gNBr,gNBs,R-MES-REP,{gNBr,PF1,PF2, {MES-RES,CODE-MES}SyKE1, T4}pk(
    gNBs));
#Claims
claim_gNBr1(gNBr,Nisynch);
claim_gNBr2(gNBr,Niagree);
claim_gNBr3(gNBr,Empty,(Fresh,T1));
claim_gNBr4(gNBr,Empty,(Fresh,T3));
  }

  role MVR
  {
  #Nonces and Variables
    var Nr: Nonce;
    fresh RAND:Nonce;
    var a5: Nonce;
    var a6: Nonce;
    fresh a7: Nonce;
  #Timestamps
    fresh T2: Timestamp;
    fresh T3: Timestamp;
    fresh T5: Timestamp;
  #Other Macro Computations
    macro CODE-MES = H(MES-ID,MES-STATE,MVR,gNBs,gNBr,RAND);  #CODE-MES
    for verification
    macro Nv1 = H(Nr,T3);          #Nonce Verifier 1
    macro PE1 = Multiply(a5,M);
    macro PF1 = Multiply(a6,M);
    macro PF2 = Multiply(a7,M);
    macro SyKF = Multiply(a6,a7,M);
    macro SyKE2 = Multiply(a5,a7,M);
  #HMAC Computations
```

```
    macro hmac2 = H(MES-ID,MES-STATE,gNBs,gNBr,Nr,PE1,PF1,MVR,T2);
    #macro hmac3 = H(MVR,MES-VER,CODE-MES,Nr,gNBr,T3);
    #macro hmac5 = H(gNBs,MES-ID,CODE-MES,T5);
#Message Specifications
recv_2(gNBr,MVR, R-MES-REQ,{MES-ID,MES-STATE,gNBs,gNBr,Nr,PE1,PF1,T2}pk(
    MVR),{H(MVR,T2)}sk(gNBr),hmac2);
match(hmac2,H(MES-ID,MES-STATE,gNBs,gNBr,Nr,PE1,PF1,MVR,T2));
send_3(MVR,gNBr, MVR-MES-REP,{MVR,PF2, {MES-VER,CODE-MES}SyKF, Nv1,T3}pk(
    gNBr),{H(gNBr,T3)}sk(MVR));
recv_5(gNBs,MVR, S-MES-VER,{gNBs,MES-ID, {CODE-MES}SyKE2, T5}pk(MVR));
#Claims
claim_MVR1(MVR,Nisynch);
claim_MVR2(MVR,Niagree);
claim_MVR3(MVR,Empty,(Fresh,T2));
claim_MVR4(MVR,Empty,(Fresh,T5));
claim_MVR5(MVR, Secret,Nr);
  }
};
```

Listing A.4: Parts - E and F Scyther Script

Scyther results : verify

Claim				Status		Comments
gNBs_gNBr_MVR	gNBs	gNBs_gNBr_MVR,gNBs1	Nisynch	Ok	Verified	No attacks.
		gNBs_gNBr_MVR,gNBs2	Niagree	Ok	Verified	No attacks.
	gNBr	gNBs_gNBr_MVR,gNBr1	Nisynch	Ok	Verified	No attacks.
		gNBs_gNBr_MVR,gNBr2	Niagree	Ok	Verified	No attacks.
	MVR	gNBs_gNBr_MVR,MVR1	Nisynch	Ok	Verified	No attacks.
		gNBs_gNBr_MVR,MVR2	Niagree	Ok	Verified	No attacks.
		gNBs_gNBr_MVR,MVR5	Secret Nr	Ok	Verified	No attacks.

Done.

Figure A.5: Scyther Validation Results for the Part E and Part F of the SP

A.3 HLPSL Scripts employed for AVISPA-based Validations and their Results

In order to confirm that proposed protocols (i.e., Part A, Part B, and Part C&D) are resilient against the attack, AVISPA tool [169] is used. AVISPA stands for Automated Validation of Internet Security Protocols and Applications. This tool offers modular and expressive formal language to specify the security features of the authentication protocols. Apart from that, it uses different types of backend server that helps carry out different types of implementation using various automatic analysis techniques ranging from protocol falsification to abstraction-based verification methods for both finite and infinite numbers of sessions. This tool uses the role-based language HLPSL (High-Level Protocols Specification Language) to model the authentication protocol in order to examine its security properties. There are four types of backend servers specified by the tool:1) On- the-fly Model-Checker (OFMC), 2) Constraint Logic-based Attack Searcher (CL-AtSe), 3) SAT-based Model-Checker (SATMC), 4) Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP).

We use the OFMC and CL-Atse backend servers to verify the proposed protocols similar to [155, 170].

A.3.1 Simulation of Part A& Part B using AVISPA tool

We do the simulation of Part A and Part B by modeling the protocol into HLPSL language. Part A protocol is modeled into the two roles $gNBSource$ and OSS, sessions, and environment in order to carry out the simulation. The same was used for Part B, two roles $gNBSource$ and TTP, session, and environment() to carry out the simulation. Listings A.5, and A.6 depict the AVISPA scripts we have formalized to achieve this validation.

```
role gNBSource(A,B:agent,
     M: text,
     F: hash_func,
```

```
    KgNBSource,Koss: public_key,
    SND_B, RCV_B: channel (dy))
played_by A
def=

  local T1, IDs, T2,T3, T4, A1: text,
  Noss, Kdos: text,
  P2: message, %% more specifically: exp(text,text)
  HMAC: hash(text.text.text.message),
  State: nat,
        X: text,
        Cert: text,
  AUTH_B: message

  const sec_a_HMAC : protocol_id

  init  State := 0

  transition

  1. State = 0  /\ RCV_B(start) =|>
     State'= 2 /\ IDs' := new()
               /\ A1' := new()
               /\ T1' := new()
          /\ SND_B( IDs'.exp(M,A1').T1' )


  2. State = 2  /\ RCV_B(IDs.P2'.Noss'.Kdos) =|>
     State'= 4 /\ T2' := new()
               /\ HMAC' := F(T1.Noss'.IDs.exp(P2',A1))
               /\ SND_B( {A.{IDs.exp(M,A1).T1.Noss'}_(inv(KgNBSource)).T2
    '}_HMAC' )


  3. State = 4  /\ RCV_B({B.{IDs.P2.Noss.T1}_(inv(Koss)).T2}_HMAC) =|>
     State'= 6 /\ X' := new()
                /\ T3' := new()
               /\ SND_B({X'.sock.T3'}_HMAC)
               /\ AUTH_B' := {IDs.P2.Noss.T1}_(inv(Koss))
```

```
                /\ secret(HMAC,sec_a_HMAC,{A,B})
                /\ witness(A,B,sk2,HMAC)

  4. State = 6  /\ RCV_B({Cert'.one.T4}_HMAC) =|>
     State':= 8 /\ request(A,B,sk1,HMAC)

end role

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

role oss (B,A:agent,
    M: text,
    F: hash_func,
    Koss, KgNBSource: public_key,
    SND_A, RCV_A: channel (dy))
played_by B
def=

  local T1, IDs, T2, T3, T4: text,
  Noss, A2, Kdos: text,
  HMAC: hash(text.text.text.message),
  P2: message,
  State: nat,
        X: text,
        Cert: text,
  AUTH_A: message

  const sec_b_HMAC : protocol_id

  init  State := 1

  transition

  1. State = 1  /\ RCV_A( IDs'.P2'.T1' ) =|>
     State':= 3 /\ A2' := new()
                /\ Noss' := new()
                /\ Kdos' := new()
                /\ SND_A(IDs'.exp(M,A2').Noss'.Kdos)
                /\ HMAC' := F(T1'.Noss'.IDs'.exp(P2',A2'))
```

```
  2. State = 3  /\ RCV_A( {A.{IDs.P2.T1.Noss}_(inv(KgNBSource)).T2'}_HMAC
    ) =|>
     State':= 5 /\ SND_A( {B.{IDs.exp(M,A2).Noss.T1}_(inv(Koss)).T2'}_HMAC
     )
                /\ AUTH_A' := {IDs.P2.T1.Noss}_(inv(KgNBSource))
                /\ witness(B,A,sk1,HMAC)
                /\ secret(HMAC,sec_b_HMAC,{A,B})

  3. State = 5  /\ RCV_A({X'.sock.T3}_HMAC) =|>
     State':= 7 /\ Cert' := new()
                 /\ T4' := new()
                /\ SND_A({Cert'.one.T4}_HMAC)
                /\ request(B,A,sk2,HMAC)

end role

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

role session(A, B: agent,
       KgNBSource, Koss: public_key,
       M: text,
             F: hash_func)
def=

  local SA, RA, SB, RB: channel (dy)

  composition

     gNBSource(A,B,M,F,KgNBSource,Koss,SA,RA)
  /\ oss(B,A,M,F,Koss,KgNBSource,SB,RB)

end role

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

role environment()
def=
```

```
  const sk1, sk2   : protocol_id,
        a, b       : agent,
        ka, kb, ki : public_key,
        g          : text,
        f          : hash_func,
        sock, one  : text

  intruder_knowledge = {g,f,a,b,ka,kb,i,ki,inv(ki),sock,one
%%%                      b,g,f,ki,kb,inv(ki)       %%% 3rd session
                        }

  composition

  session(a,b,ka,kb,g,f)
     /\ session(a,i,ka,ki,g,f)
     /\ session(i,b,ki,kb,g,f)

end role

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

goal

  %secrecy_of HMAC
  secrecy_of sec_a_HMAC, sec_b_HMAC % Addresses M9

  %Alice authenticates Bob on sk1
  authentication_on sk1 % Addresses M1, M2, M3, M7, M10
  %Bob authenticates Alice on sk2
  authentication_on sk2 % Addresses M1, M2, M3, M7, M10

end goal

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

environment()
```

Listing A.5: Part-A AVISPA Script

```

```

```
HLPSL:
role gNBSource(A,B:agent,
     M: text,
     F: hash_func,
     KgNBSource,Kttp: public_key,
     SND_B, RCV_B: channel (dy))
played_by A
def=

  local T1, IDs, T2,T3, T4, A1: text,
  Nttp, Kdos: text,
  P2: message, %% more specifically: exp(text,text)
  HMAC: hash(text.text.text.message),
  State: nat,
        X: text,
        Cert: text,
  AUTH_B: message

  const sec_a_HMAC : protocol_id

  init  State := 0

  transition

  1. State = 0  /\ RCV_B(start) =|>
     State':= 2 /\ IDs' := new()
                /\ A1' := new()
                /\ T1' := new()
          /\ SND_B( IDs'.exp(M,A1').T1' )


  2. State = 2  /\ RCV_B(IDs.P2'.Nttp'.Kdos) =|>
     State':= 4 /\ T2' := new()
                /\ HMAC' := F(T1.Nttp'.IDs.exp(P2',A1))
                /\ SND_B( {A.{IDs.exp(M,A1).T1.Nttp'}_(inv(KgNBSource)).T2
    '}_HMAC' )


  3. State = 4  /\ RCV_B({B.{IDs.P2.Nttp.T1}_(inv(Kttp)).T2}_HMAC) =|>
```

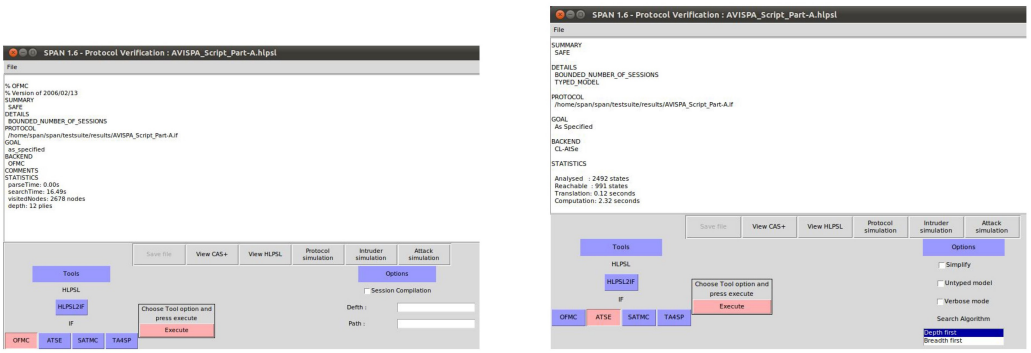
```
     State'= 6 /\ X' := new()
41                 /\ T3' := new()
                   /\ SND_B({X'.sock.T3'}_HMAC)
                   /\ AUTH_B' := {IDs.P2.Nttp.T1}_(inv(Kttp))
                   /\ secret(HMAC,sec_a_HMAC,{A,B})
                   /\ witness(A,B,sk2,HMAC)
46
  4. State = 6  /\ RCV_B({Cert'.one.T4}_HMAC) =|>
     State'= 8 /\ request(A,B,sk1,HMAC)

end role
51
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

role ttp (B,A:agent,
    M: text,
56  F: hash_func,
    Kttp, KgNBSource: public_key,
    SND_A, RCV_A: channel (dy))
played_by B
def=
61
  local T1, IDs, T2, T3, T4: text,
  Nttp, A2, Kdos: text,
  HMAC: hash(text.text.text.message),
  P2: message,
66 State: nat,
       X: text,
       Cert: text,
  AUTH_A: message

71 const sec_b_HMAC : protocol_id

  init  State := 1

  transition
76
  1. State = 1  /\ RCV_A( IDs'.P2'.T1' ) =|>
     State'= 3 /\ A2' := new()
```

```
                /\ Nttp' := new()
                /\ Kdos' := new()
81              /\ SND_A(IDs'.exp(M,A2').Nttp'.Kdos)
                /\ HMAC' := F(T1'.Nttp'.IDs'.exp(P2',A2'))

  2. State = 3  /\ RCV_A( {A.{IDs.P2.T1.Nttp}_(inv(KgNBSource)).T2'}_HMAC
    ) =|>
     State':= 5 /\ SND_A( {B.{IDs.exp(M,A2).Nttp.T1}_(inv(Kttp)).T2'}_HMAC
     )
86              /\ AUTH_A' := {IDs.P2.T1.Nttp}_(inv(KgNBSource))
                /\ witness(B,A,sk1,HMAC)
                /\ secret(HMAC,sec_b_HMAC,{A,B})

  3. State = 5  /\ RCV_A({X'.sock.T3}_HMAC) =|>
91    State':= 7 /\ Cert' := new()
                  /\ T4' := new()
                /\ SND_A({Cert'.one.T4}_HMAC)
                /\ request(B,A,sk2,HMAC)

96 end role

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

role session(A, B: agent,
101      KgNBSource, Kttp: public_key,
       M: text,
             F: hash_func)
def=

106  local SA, RA, SB, RB: channel (dy)

  composition

     gNBSource(A,B,M,F,KgNBSource,Kttp,SA,RA)
111  /\ ttp(B,A,M,F,Kttp,KgNBSource,SB,RB)

end role

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
116
role environment()
def=

  const sk1, sk2   : protocol_id,
121       a, b       : agent,
        ka, kb, ki : public_key,
        g          : text,
        f          : hash_func,
        sock, one  : text
126
  intruder_knowledge = {g,f,a,b,ka,kb,i,ki,inv(ki),sock,one
%%%                      b,g,f,ki,kb,inv(ki)       %%% 3rd session
                        }

131  composition

  session(a,b,ka,kb,g,f)
     /\ session(a,i,ka,ki,g,f)
     /\ session(i,b,ki,kb,g,f)
136
end role

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

141 goal

  %secrecy_of HMAC
  secrecy_of sec_a_HMAC, sec_b_HMAC % Addresses M9

146  %Alice authenticates Bob on sk1
  authentication_on sk1 % Addresses M1, M2, M3, M7, M10
  %Bob authenticates Alice on sk2
  authentication_on sk2 % Addresses M1, M2, M3, M7, M10

151 end goal

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

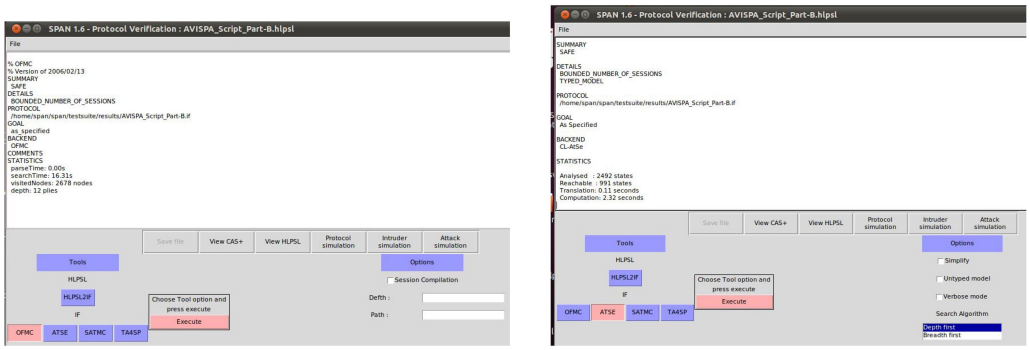
```
environment()
```

Listing A.6: Part-B AVISPA Script



(a) (b)

Figure A.6: AVISPA outcome for Part A using (a) OFMC backend server (b) CL-Atse backend server.



(a) (b)

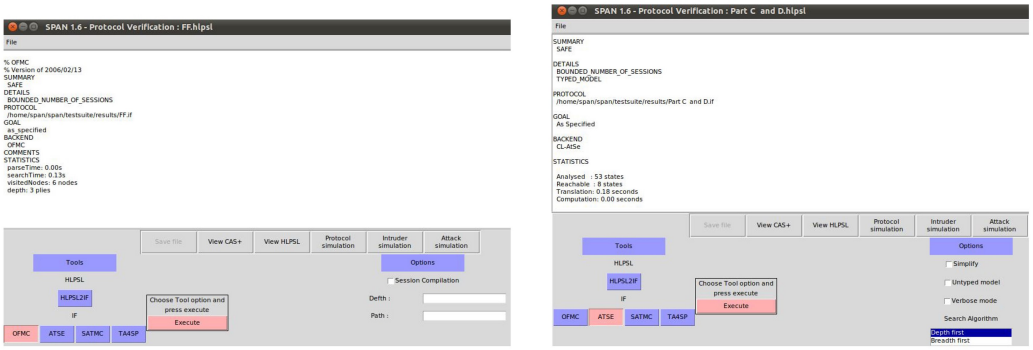
Figure A.7: AVISPA outcome for Part B using (a) OFMC backend server (b) CL-Atse backend server.

Fig A.6-(a), Fig A.6-(b), Fig A.7-(a), and Fig A.7-(b) show that the protocol is safe and secure.

A.3.2 Simulation of Part C&D using AVISPA tool

We do the simulation of Part C&D by modeling the protocol into HLPSL language. In Part C&D protocol is modeled into the three roles gNB_{Source},

$gNB_{Roaming}$, and OSS, sessions, and environment in order to carry out the simulation. Fig A.8-(a) and Fig A.8-(b) show that the protocol is safe and secure.



(a) (b)

Figure A.8: AVISPA outcome for Part C using (a) OFMC backend server (b) CL-Atse backend server.

A.4 DoS Puzzle

The formulated DoS puzzle: $H[ID_S||ID_C||n_S||n_C||X] = 0_1 0_2 0_3 ... 0_{k_{DoS}} Y$ was implemented within the prototype protocol as a java function. Listing A.7 represents the code fragments coded to solve the puzzle and determine X (i.e. DoS_Puzzle()), while DoS_Puzzle_Verification() function indicates whether the received X is the actual solution to the Puzzle.

```
public String DoS_Puzzle(int k_dos, String PublicKey,  String Client_ID,
    String Server_ID, String Client_Nonce, String Server_Nonce) throws
    UnknownHostException, Exception{
long j = 0;
String X;
System.out.println("\n\nDoS PUZZLE STARTING.....");
long ts_start = System.currentTimeMillis();
    while (true){
        X = RandomNonceGenerator();
        String BIhash = BIHash(PublicKey + Client_ID + Client_Nonce +
    Server_ID + Server_Nonce + X);
```

```
            if (CheckZeroCount(BIhash,k_dos)==true){
                System.out.println("SOLUTION FOUND....X = "+X);
                break;
            }
            j++;
    }
        long ts_end = System.currentTimeMillis();
        System.out.println("Number of Attempts :"+j);
        System.out.println("DoS Puzzle Process Time [ms]:"+(ts_end-
    ts_start));
        return X;
}
public void DoS_Puzzle_Verification(int k_dos, String PublicKey,  String
    Client_ID, String Server_ID, String Client_Nonce, String Server_Nonce,
     String X) throws UnknownHostException, Exception{
String VerifyingHash = BIHash(PublicKey + Client_ID + Client_Nonce +
    Server_ID + Server_Nonce + X);
    System.out.println("Received X : " + X);
    System.out.println("Verifying Hash : " + VerifyingHash);
            if (CheckZeroCount(VerifyingHash,k_dos)){
                System.out.println(".................The DoS Puzzle is
    VERIFIED.............\n\n");
            }else {
                System.out.println(".............The DoS Puzzle is Not
    Verified ==> DoS Attack Detected.........\n\n");
            }
}
```

Listing A.7: DoS Puzzle and Verification

Further, Listing A.8 specifies two other functions employed within the DoS puzzle functions mentioned above.

```
public static String BIHash (String message) throws
    NoSuchAlgorithmException {
String HashAlgorithm = "SHA-512";
MessageDigest md = MessageDigest.getInstance(HashAlgorithm);
int k = 155;
byte[] messageDigest = md.digest(message.getBytes());
BigInteger no = new BigInteger(1, messageDigest);
int BIlength = no.toString().length();
```

```
String hash = no.toString();
    if( BIlength < k ){
        for(int i=0; i < (k-BIlength); i++){
            hash = "0"+hash;
        }
    }
    return  hash;
}
public static Boolean CheckZeroCount(String hash, int k_dos){
    char[] hashArray = hash.toCharArray();
    Boolean Check = false;
    for(int i=0; i < k_dos ; i++){
        if (hashArray[i] == '0'){
            Check = true;
        }else {
            Check = false;
            break;
        }
    }
    return Check;
}
```

Listing A.8: BIHash and CheckZeroCount Functions

Table A.2: The Solving Time of the DoS Puzzle

Complexity	No. of Hashing Attempts	Completion Time [ms]
1	1	0.5
2	13	2
3	128	12
4	1398	70
5	13615	588
6	100325	4300
7	1334136	58655

The k_{DoS} parameter is an important value for solving the presented puzzle. In fact, this value determines the complexity of the puzzle, which controls the solving time of the puzzle. TABLE A.2 specify the solving times of the puzzle for considered complexity. It is evident that $k_{DoS} = 4$ is a suitable value for the

proposed security protocol.

A.5 Protocol Details of the Legacy Protocol

In order to evaluate the effectiveness of the embedded security measures of the proposed SP that targeted freshness, integrity, and DoS measures, we defined a Legacy Protocol (LP) detaching additional security mechanisms of the SP. Following Fig. A.9,A.10,A.11,A.12, and A.13 illustrate the alterations conducted to form the LP.

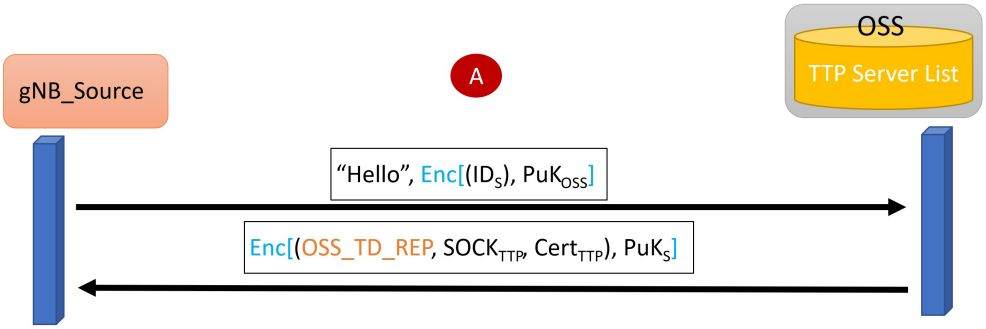


Figure A.9: Part A of the Legacy Protocol

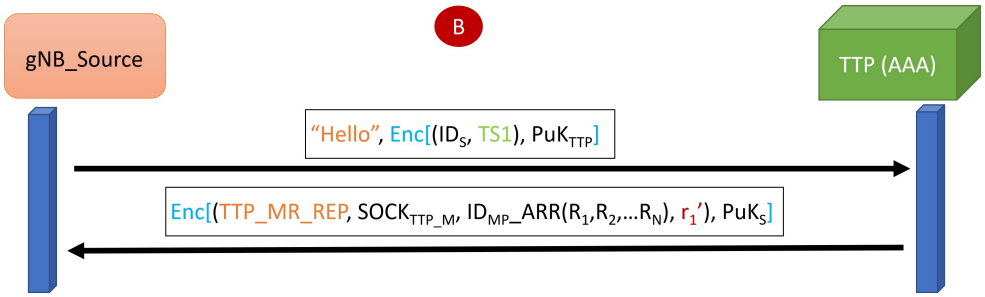


Figure A.10: Part B of the Legacy Protocol

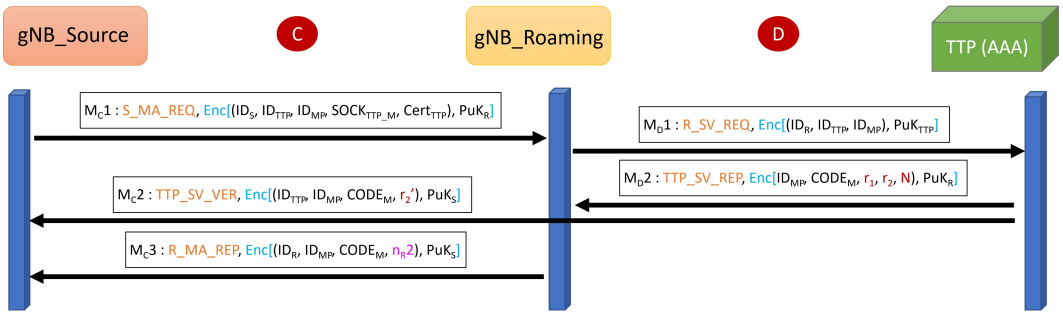


Figure A.11: Part C and D of the Legacy Protocol

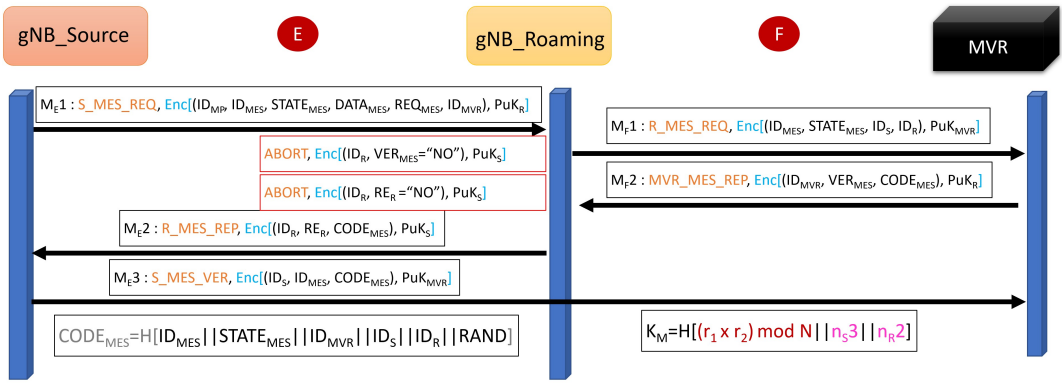


Figure A.12: Part E and F of the Legacy Protocol

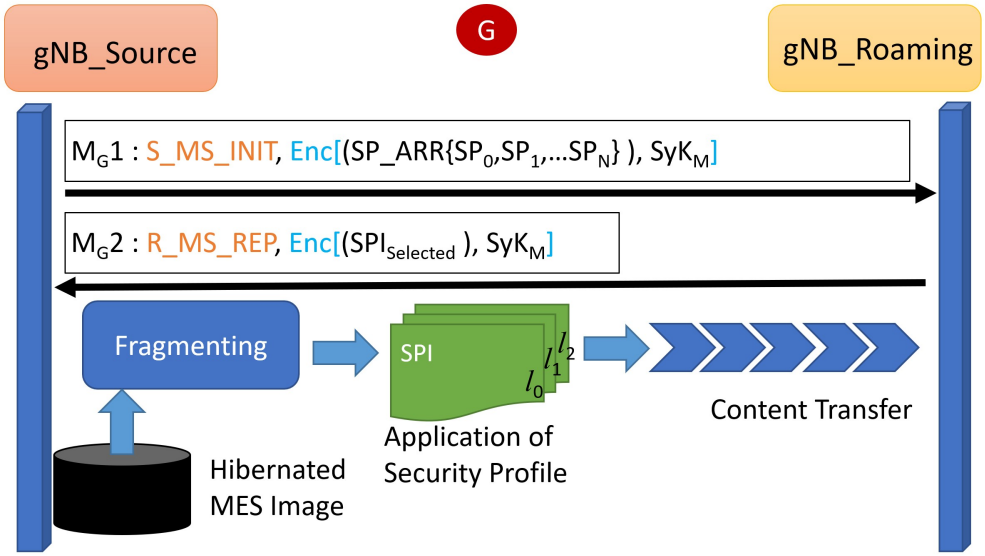


Figure A.13: Part G of the Legacy Protocol

A.6 MatLab Code for the Simulation in Fig. 5.16

In Chapter 5, Fig. 5.16 illustrates the behaviors of the SP and the LP in case of a tampering occurrence, specified under different tampering probabilities. This simulation was formulated with Matlab taking actual timing values of the developed prototype protocol. The re-transmission delays were computed by taking the Encryption Processing Delay (EPD), Transmission Delay (TD), and Decryption Processing Delay (DPD) components for each message. The formulated model is specified in Listing A.9.

```
N_trials = 400;
S = 50;
Probability = (1/S):(1/S):1.0;
CT = zeros(3,S); %Completion time
CT_L = zeros(3,S); %Completion time Legacy Scenario
Total_Completion_Time = zeros;
Total_Completion_Time_L = zeros;

%Proposed Standard Protocol
%There are 20 messages (A -> 5, B -> 5, C -> 3, D -> 2, E -> 3, F -> 2, G
    ->2)
Messages = 1:1:20;
EPD = [22.0292 23.0939 4.0767 63.5275 20.4762 23.3527 3.7874 87.1718
    19.5384 19.9243 78.1628 58 586.5213 0.9112 20.1910 83.3304 56.9223
    57.4855 144.7541 0.1135];
TD = [25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25];
DPD = [22.2593 19.2160 22.8990 76.9597 20.1105 17.6348 20.5719 77.2448
    18.6869 18.6901 76.2258 72.3221 72.5634 17.2479 20.4039 25.5 73.2724
    80.4547 55.7157 275.6086];

%Protocol Completion Time
Delay_A = EPD(1)+TD(1)+DPD(1)+EPD(2)+TD(2)+DPD(2)+EPD(3)+TD(3)+DPD(3)+EPD
    (4)+TD(4)+DPD(4);
Delay_B = EPD(5)+TD(5)+DPD(5)+EPD(6)+TD(6)+DPD(6)+EPD(7)+TD(7)+DPD(7)+EPD
    (8)+TD(8)+DPD(8);
Delay_CD = EPD(9)+TD(9)+DPD(9)+EPD(10)+TD(10)+DPD(10)+EPD(11)+TD(11)+DPD
    (11)+EPD(13)+TD(13)+DPD(13);
Delay_EF = EPD(14)+TD(14)+DPD(14)+EPD(15)+TD(15)+DPD(15)+EPD(16)+TD(16)+
    DPD(16)+EPD(17)+TD(17)+DPD(17)+EPD(18);
```

```
Delay_G = EPD(19)+TD(19)+DPD(19)+EPD(20)+TD(20)+DPD(20);

Total_Delay = Delay_A + Delay_B + Delay_CD + Delay_EF + Delay_G;
disp(Total_Delay);

%Legacy Protocol
%There are 16 messages (A -> 2, B -> 2, C -> 3, D -> 2, E -> 3, F -> 2, G
    ->2)
EPD_L = [22 22 22 22 19.5384 19.9243 54.1628 58 200.5213 0.9112 20.1910
    83.3304 56.9223 57.4855 50.7541 0.1135];
TD_L = [25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25];
DPD_L = [22 22 22 22 18.6869 18.6901 76.2258 72.3221 72.5634 17.2479
    20.4039 25.5 73.2724 80.4547 55.7157 275.6086];

%Protocol Completion Time - Legacy
L_Delay_A = EPD_L(1)+TD_L(1)+DPD_L(1)+EPD_L(2)+TD_L(2)+DPD_L(2);
L_Delay_B = EPD_L(3)+TD_L(3)+DPD_L(3)+EPD_L(4)+TD_L(4)+DPD_L(4);
L_Delay_CD = EPD_L(5)+TD_L(5)+DPD_L(5)+EPD_L(6)+TD_L(6)+DPD_L(6)+EPD_L(7)+
    TD_L(7)+DPD_L(7)+EPD_L(9)+TD_L(9)+DPD_L(9);
L_Delay_EF = EPD_L(10)+TD_L(10)+DPD_L(10)+EPD_L(11)+TD_L(11)+DPD_L(11)+
    EPD_L(12)+TD_L(12)+DPD_L(12)+EPD_L(13)+TD_L(13)+DPD_L(13)+EPD_L(14);
L_Delay_G = EPD_L(15)+TD_L(15)+DPD_L(15)+EPD_L(16)+TD_L(16)+DPD_L(16);

Total_Delay_L = L_Delay_A + L_Delay_B + L_Delay_CD + L_Delay_EF +
    L_Delay_G;
disp(Total_Delay_L);

% For Standard Protocol
disp('Standard Protocol');
disp('=================');
for P = 1:1:S

    disp('Trial');
    disp(P);
    disp('Probability');
    disp(Probability(P));
    N_tampered = N_trials * Probability(P);

    for y = 1:1:N_trials
```

```
    Tampered_Messages = randi([0,N_trials], 1, floor(N_tampered));
    disp('Tampered Messages');
    disp(Tampered_Messages);
    %Assuming a tempering is detected within the decryption processing
    interval
    Tampered_Message_No = rem(Tampered_Messages,20);
    disp('Tampered Message Numbers');
    disp(Tampered_Message_No);

    Tampered_Message_No = nonzeros(Tampered_Message_No);
    Retransmission_Delay = zeros;
        for x = Tampered_Message_No
           Retransmission_Delay(x) = EPD(x)+TD(x)+DPD(x);
        end
    disp('Retransmission Delay');
    disp(Retransmission_Delay);

    Total_Completion_Time(y) = Total_Delay + sum(Retransmission_Delay);
    end

    disp('Total Completion Time');
    disp(Total_Completion_Time);
    CI95 = [mean(Total_Completion_Time) - 1.96*(std(Total_Completion_Time)
    ./sqrt(numel(Total_Completion_Time))), mean(Total_Completion_Time) +
    1.96*(std(Total_Completion_Time)./sqrt(numel(Total_Completion_Time)))
    ];
    mu = mean(Total_Completion_Time);
    CT(1,P) = max(CI95)- mu;
    CT(2,P) = mu - min(CI95);
    CT(3,P) = mu;
    disp('CI95 Maximum :');
    disp(CT(1,P));
    disp('CI95 Minimum :');
    disp(CT(2,P));
    disp('Mean :');
    disp(CT(3,P));
end

% For Legacy Protocol
```

```
disp('Legacy Protocol');
disp('===============');

for P = 1:1:S
    disp('Trial');
    disp(P);
    disp('Probability');
    disp(Probability(P));
    N_tampered = N_trials * Probability(P);

    for y = 1:1:N_trials
    Tampered_Messages = randi([0,N_trials], 1, floor(N_tampered));
    disp('Tampered Messages');
    disp(Tampered_Messages);
    %Assuming a tempering is detected within the decryption processing
    interval
    Tampered_Message_No = rem(Tampered_Messages,20);
    disp('Tampered Message Numbers');
    disp(Tampered_Message_No);
    Tampered_Message_No = nonzeros(Tampered_Message_No);
    %disp(Tampered_Message_No);
    Retransmission_Delay = zeros;

        for x = Tampered_Message_No
            if x < 3
                Retransmission_Delay(x) = L_Delay_A+EPD_L(3)+TD_L(3)+DPD_L
    (3);
            elseif x < 5
                Retransmission_Delay(x) =  L_Delay_B + EPD_L(5)+TD_L(5)+
    DPD_L(5);
            elseif x < 10
                Retransmission_Delay(x) = EPD_L(5)+TD_L(5)+DPD_L(5)+EPD_L
    (6)+TD_L(6)+DPD_L(6)+EPD_L(7)+TD_L(7)+DPD_L(7);
            elseif x < 16
                Retransmission_Delay(x) = EPD_L(10)+TD_L(10)+DPD_L(10)+
    EPD_L(11)+TD_L(11)+DPD_L(11)+EPD_L(12)+TD_L(12)+DPD_L(12);
            else
                Retransmission_Delay(x) = EPD_L(15)+TD_L(15)+DPD_L(15)+
    EPD_L(16)+TD_L(16)+DPD_L(16);
```

```
            end
        end
    disp('Retransmission Delay');
    disp(Retransmission_Delay);
    Total_Completion_Time_L(y) = Total_Delay_L + sum(Retransmission_Delay)
    ;
    end

    disp('Total Completion Time');
    disp(Total_Completion_Time_L);
    CI95_L = [mean(Total_Completion_Time_L) - 1.96*(std(
    Total_Completion_Time_L)./sqrt(numel(Total_Completion_Time_L))), mean(
    Total_Completion_Time_L) + 1.96*(std(Total_Completion_Time_L)./sqrt(
    numel(Total_Completion_Time_L)))];
    mu_L = mean(Total_Completion_Time_L);
    CT_L(1,P) = max(CI95_L) - mu_L;
    CT_L(2,P) = mu_L - min(CI95_L);
    CT_L(3,P) = mu_L;
    disp('CI95 Maximum :');
    disp(CT_L(1,P));
    disp('CI95 Minimum :');
    disp(CT_L(2,P));
    disp('CI95 Mean :');
    disp(CT_L(3,P));
end

disp(CT);
disp(CT_L);

errorbar(Probability,CT(3,:),CT(1,:),'-s','MarkerSize',15,'MarkerEdgeColor
    ','b','MarkerFaceColor','c'); hold on;
errorbar(Probability,CT_L(3,:),CT_L(1,:),'-o','MarkerSize',15,'
    MarkerEdgeColor','r','MarkerFaceColor','g')
```

Listing A.9: MatLAB Code for Tampering Probability Simulations

Though the graph was plotted as an error bar graph that indicates the Confidence Interval (CI) of 95%, the CI range is quite difficult to visualize within the given time frame.

A.7 Details of the MEC Prototype Development Environment

As explicated in the paper, the interfaces of each entity involved in this proposed protocol were implemented as java files that establish connections through an IPC-based approach. Due to the larger file sizes and repetitive coding blocks, we shared the program files in the following GitHub repository: `https://github.com/Pasika87/MSMAP`. All the program files and the corresponding private and public keys required for the execution are included in the repository. The README file is providing the required instructions to execute the program. This program is written to embed a Command Line Interface (CLI) as the I/O terminal. Thus, all the protocol steps, outcomes, transferring messages (encrypted, or otherwise), keys, Security Profiles, timing values, and all other security measures are displayed via the CLI for the users' reference. The following screenshots account for such outcomes in the program execution via the CLI (i.e. Fig. B.2, B.3,B.4, and B.5). Though these screenshots were extracted during the localhost operation, the experiments conducted in the paper were solely based on the experimental MEC environment specified under Section VII of the paper.

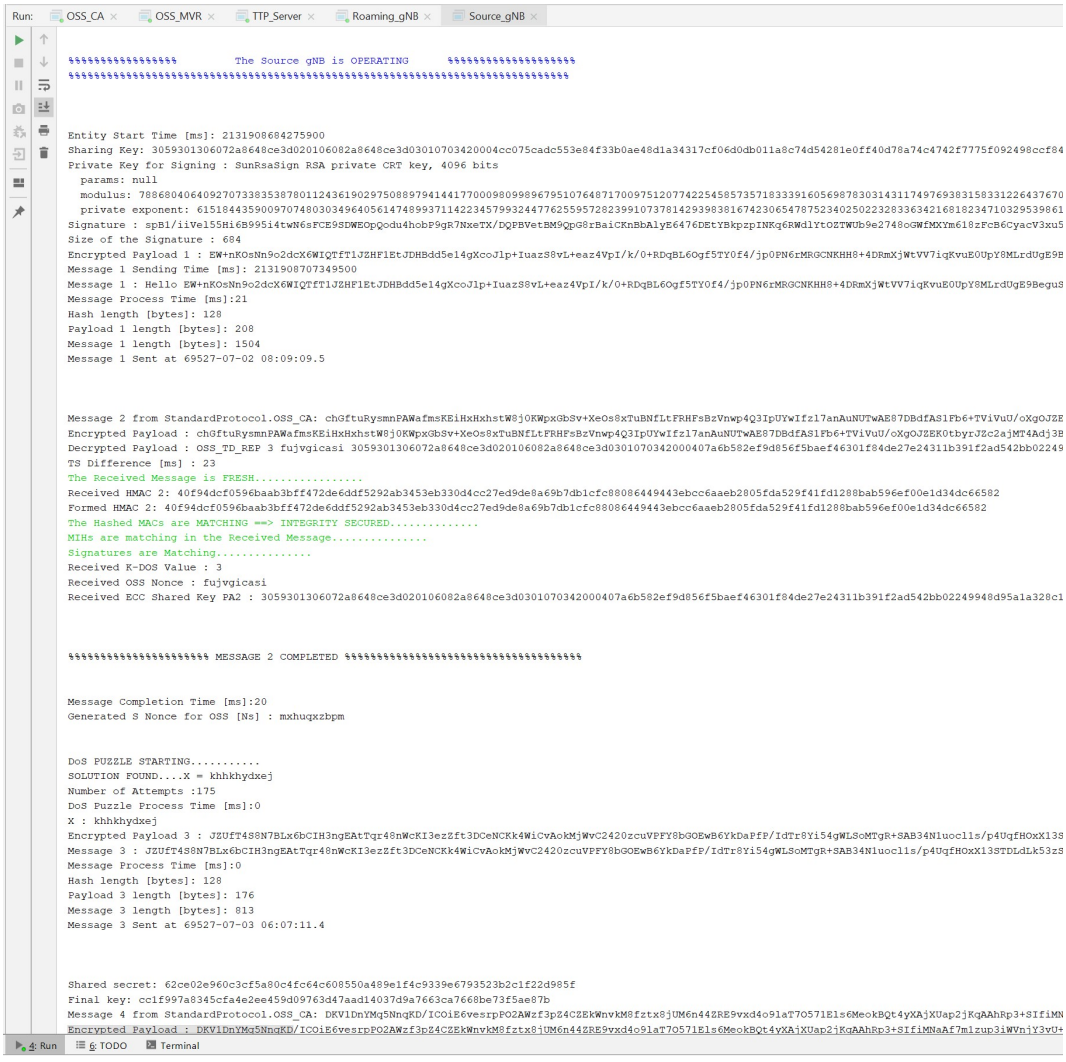


Figure A.14: CLI Outcome 1

Figure A.15: CLI Outcome 2

Figure A.16: CLI Outcome 3

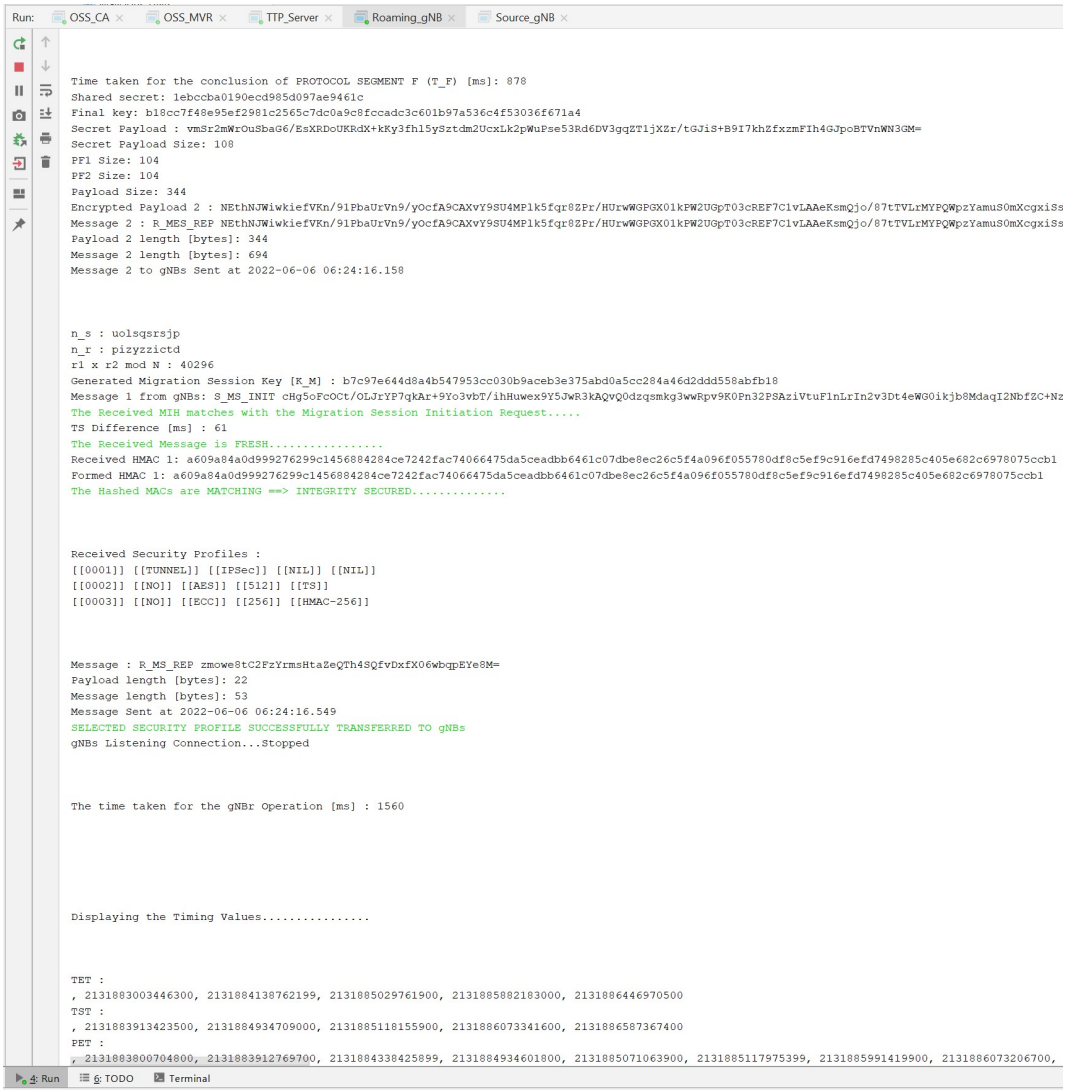


Figure A.17: CLI Outcome 4

APPENDIX

TWO

SUPPLEMENTARY APPENDICES FOR CHAPTER 6: MEC SERVICE MIGRATION SECURITY MANAGEMENT MODEL

This appendix presents the implementations related to Chapter 5, in regard to presented simulations and emulations. This document contains the following.

1. Appendix B.1: AES Cryptographic Digest/Overhead Computing Scheme
2. Appendix B.2: AES Cryptographic Digest/Overhead Variation Plot
3. Appendix B.3: Cost Computation Program for Security Algorithms
4. Appendix B.4: Simulating Migration Time Variation for Different Security Settings/ Algorithms using MatLab
5. Appendix B.5: Prototype MEC SMSF Implementation

B.1 AES Cryptographic Digest/Overhead Computing Scheme

Listing B.1 depicts the algorithm for computing the cryptographic digest of the AES algorithm. Once the input I is specified, the outcome will include the

digest size in bits for different padding schemes and block cipher modes. This computation is relevant for the AES cryptographic digest model presented in Fig. 6.3.

```
AES_K_1 = 128; %bits
AES_K_2 = 192; %bits
AES_K_3 = 256; %bits
Padding_Scheme_1 = "PKCS5Padding";
Padding_Scheme_2 = "NoPadding";
Block_Cipher_Mode_1 = "CBC";
Block_Cipher_Mode_2 = "ECB";
Block_Cipher_Mode_3 = "CTR";

I = 100000; %Input Size

%PKCS5Padding and CBC/ ECB

disp(Padding_Scheme_1);
disp(Block_Cipher_Mode_1);
disp(Block_Cipher_Mode_2);

O_min = 24;
delta = 20;
divisor = 48;

O_max = (O_min + 2*delta);

if (I < 48)
    I_hat = 0;
    I_dash = I;
else
    I_hat = fix(I/divisor);
    I_dash = rem(I,divisor);
end

O = O_max * I_hat;

if (I_dash < 16)
    O = O + O_min;
elseif (I_dash < 32)
```

```
    O = O + O_min + delta;
else
    O = O + O_min + 2 * delta;
end

disp('The Input Size : ');
disp(I);
disp('The Output Size : ');
disp(O);


%NoPadding and CBC

disp(Padding_Scheme_2);
disp(Block_Cipher_Mode_1);
disp(Block_Cipher_Mode_2);

O_min = 24;
delta = 20;
divisor = 48;

if (rem(I,16) == 0)

O_max = (O_min + 2*delta);

if (I < 64)
    I_hat = 0;
    I_dash = I;
else
    I_hat = fix(I/divisor);
    I_dash = rem(I,divisor);
end

O = O_max * I_hat;

if (I_dash == 16)
    O = O + O_min;
elseif (I_dash == 32)
    O = O + O_min + delta;
```

```
else
    O = O + O_min + 2 * delta;
end

disp('The Input Size : ');
disp(I);
disp('The Output Size : ');
disp(O);

else
    disp('The Input Size is not divisable by 16: Cannot Perform Encryption
    ..... ');
end


%PKCS5Padding & NoPadding for CTR

disp(Padding_Scheme_1);
disp(Padding_Scheme_2);
disp(Block_Cipher_Mode_3);

O_min = 4;
alpha = 4;
divisor = 3;

I_hat = round(I/divisor);

O = I_hat * O_min;

disp('The Input Size : ');
disp(I);
disp('The Output Size : ');
disp(O);
```

Listing B.1: MatLAB Code for AES Cost Computation

B.2 AES Cryptographic Digest/Overhead Variation Plot

Listing B.2 depicts the code for plotting the cryptographic digest of the AES algorithm for the computations corresponding to Appendix B.1. This code is relevant for the AES cryptographic digest plot presented in Fig. 6.4.

```
AES_K_1 = 128; %bits
AES_K_2 = 192; %bits
AES_K_3 = 256; %bits
Padding_Scheme_1 = "PKCS5Padding";
Padding_Scheme_2 = "NoPadding";
Block_Cipher_Mode_1 = "CBC";
Block_Cipher_Mode_2 = "ECB";
Block_Cipher_Mode_3 = "CTR";

End_Size = 10^4;

I_1 = 8:8:(8*(End_Size)); %Input Size Scheme 1
I_2 = 16:16:(8*(End_Size)); %Input Size Scheme 2
I_3 = 8:8:(8*(End_Size)); %Input Size Scheme 3

theta_1 = 8:8:(8*(End_Size));
theta_2 = 16:16:(8*(End_Size));
theta_3 = 8:8:(8*(End_Size));

%PKCS5Padding and CBC/ ECB

O_min = 24;
delta = 20;
divisor = 48;

O_max = (O_min + 2*delta);

I_1_length = length(I_1);
i = 1;

while (i <= I_1_length)
```

```
    if (I_1(i) < 48)
        I_hat = 0;
        I_dash = I_1(i);
    else
        I_hat = fix(I_1(i)/divisor);
        I_dash = rem(I_1(i),divisor);
    end

    O = O_max * I_hat;

    if (I_dash < 16)
        O = O + O_min;
    elseif (I_dash < 32)
        O = O + O_min + delta;
    else
        O = O + O_min + 2 * delta;
    end

    theta_1(i) = O;

    i = i+1;
end


%NoPadding and CBC

O_min = 24;
delta = 20;
divisor = 48;

O_max = (O_min + 2*delta);

I_2_length = length(I_2);
i = 1;

while (i <= I_2_length)

if (I_2(i) < 64)
    I_hat = 0;
```

```
    I_dash = I_2(i);
else
    I_hat = fix(I_2(i)/divisor);
    I_dash = rem(I_2(i),divisor);
end

O = O_max * I_hat;

if (I_dash == 16)
    O = O + O_min;
elseif (I_dash == 32)
    O = O + O_min + delta;
else
    O = O + O_min + 2 * delta;
end

theta_2(i) = O;

i = i+1;

end


%PKCS5Padding & NoPadding for CTR

O_min = 4;
divisor = 3;

I_3_length = length(I_3);
i = 1;

while (i <= I_3_length)

I_hat = round(I_3(i)/divisor);

O = I_hat * O_min;

theta_3(i) = O;
```

```
i = i+1;

end

plot(I_1,theta_1)
hold on
plot(I_2,theta_2)
hold on
plot(I_3,theta_3)
hold off
```

Listing B.2: MatLAB Code for AES Digest Variation Plot

B.3 Cost Computation Program for Security Algorithms

Listing B.3 depicts the code for computing t_E, t_D, and θ values of the security algorithms AES (i.e. key sizes of 128b,192b, and 256), BF, and RC4. The algorithms and security mechanisms are specified in the SecurityMechanisms class depicted in the Listing B.4. The results of this computation are depicted under Fig. 6.5, Fig. 6.6, and Fig. 6.7.

```
package MigrationModel;

import javax.crypto.BadPaddingException;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import java.io.IOException;
import java.nio.charset.Charset;
import java.nio.charset.StandardCharsets;
import java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.sql.SQLOutput;
import java.util.Random;

import static MigrationModel.MigrationTransfer.TimeDifference;
```

```
public class SecurityAlgorithmCostComputation {
    public static long PlaintextLength = 230*1024*1024;
    public static int Runs = 10;
    public static String SECRET_KEY_RC4 = "ThisIsTheRC4SecretKey";
    public static long EncryptionTime[][];
    public static long DecryptionTime[][];
    public static long theta[][];
    public static long AverageEncryptionTime[];
    public static long AverageDecryptionTime[];
    public static long Averagetheta[];

    public static void main(String[] args) throws IOException,
    InvalidKeyException, IllegalBlockSizeException, BadPaddingException,
    NoSuchAlgorithmException, NoSuchPaddingException,
    InvalidAlgorithmParameterException {
        EncryptionTime = new long[100][100];
        DecryptionTime = new long[100][100];
        theta = new long[100][100];
        AverageEncryptionTime = new long[100];
        AverageDecryptionTime = new long[100];
        Averagetheta = new long[100];
        int x = 0;
        while(x < (Runs+2)){
            System.out.println("Run : "+x);
            CostCalculation(x);
            x++;
        }

        //Computing Average Encryption Time
        //AES-128
        long total = 0;
        for (int i=2; i < (Runs+2); i++){
            total = total + EncryptionTime[0][i];
        }
        AverageEncryptionTime[0] = total/Runs;
        //AES-192
        total = 0;
```

```
        for (int i=2; i < (Runs+2); i++){
            total = total + EncryptionTime[1][i];
        }
        AverageEncryptionTime[1] = total/Runs;
        //AES-256
        total = 0;
        for (int i=2; i < (Runs+2); i++){
            total = total + EncryptionTime[2][i];
        }
        AverageEncryptionTime[2] = total/Runs;
        //RC4
        total = 0;
        for (int i=2; i < (Runs+2); i++){
            total = total + EncryptionTime[3][i];
        }
        AverageEncryptionTime[3] = total/Runs;
        //BF
        total = 0;
        for (int i=2; i < (Runs+2); i++){
            total = total + EncryptionTime[4][i];
        }
        AverageEncryptionTime[4] = total/Runs;

        //Computing Average Decryption Time
        //AES-128
        total = 0;
        for (int i=2; i < (Runs+2); i++){
            total = total + DecryptionTime[0][i];
        }
        AverageDecryptionTime[0] = total/Runs;
        //AES-192
        total = 0;
        for (int i=2; i < (Runs+2); i++){
            total = total + DecryptionTime[1][i];
        }
        AverageDecryptionTime[1] = total/Runs;
        //AES-256
        total = 0;
        for (int i=2; i < (Runs+2); i++){
```

```
            total = total + DecryptionTime[2][i];
        }
        AverageDecryptionTime[2] = total/Runs;
        //RC4
        total = 0;
        for (int i=2; i < (Runs+2); i++){
            total = total + DecryptionTime[3][i];
        }
        AverageDecryptionTime[3] = total/Runs;
        //BF
        total = 0;
        for (int i=2; i < (Runs+2); i++){
            total = total + DecryptionTime[4][i];
        }
        AverageDecryptionTime[4] = total/Runs;
        //Computing Average THETA
        //AES-128
        long tot = 0;
        for (int i=2; i < (Runs+2); i++){
            tot = tot + theta[0][i];
        }
        Averagetheta[0] = tot/Runs;
        //AES-192
        tot = 0;
        for (int i=2; i < (Runs+2); i++){
            tot = tot + theta[1][i];
        }
        Averagetheta[1] = tot/Runs;
        //AES-256
        tot = 0;
        for (int i=2; i < (Runs+2); i++){
          tot = tot + theta[2][i];
        }
        Averagetheta[2] = tot/Runs;
        //RC4
        tot = 0;
        for (int i=2; i < (Runs+2); i++){
            tot = tot + theta[3][i];
        }
```

```
        Averagetheta[3] = tot/Runs;
        //BF
        tot = 0;
        for (int i=2; i < (Runs+2); i++){
            tot = tot + theta[4][i];
        }
        Averagetheta[4] = tot/Runs;
        //Printing Average Values
        VerticalSpace();
        System.out.println("Encryption Times : ");
        for (int j = 0; j < 5; j++){
            System.out.println(AverageEncryptionTime[j]);
        }
        VerticalSpace();
        System.out.println("Decryption Times : ");
        for (int j = 0; j < 5; j++){
            System.out.println(AverageDecryptionTime[j]);
        }
        VerticalSpace();
        System.out.println("Theta Values : ");
        for (int j = 0; j < 5; j++){
            System.out.println(Averagetheta[j]);
        }
    }

    public static void CostCalculation(int x) throws IOException,
    InvalidKeyException, IllegalBlockSizeException, BadPaddingException,
    NoSuchAlgorithmException, NoSuchPaddingException,
    InvalidAlgorithmParameterException{
        String Plaintext = RandomStringGenerator();

        long start_time = System.nanoTime();
        String RC4EncryptedText = SecurityMechanisms.RC4Encrypt(Plaintext,
    SECRET_KEY_RC4.getBytes());
        long end_time = System.nanoTime();
        long RC4_Te = TimeDifference(start_time,end_time)/1000000;
        long RC4_theta = RC4EncryptedText.length();
        start_time = System.nanoTime();
        String BFEncryptedText = SecurityMechanisms.BF_encrypt(Plaintext,
```

```
SECRET_KEY_RC4);
    end_time = System.nanoTime();
    long BF_Te = TimeDifference(start_time,end_time)/1000000;
    long BF_theta = BFEncryptedText.length();
    start_time = System.nanoTime();
    String AESEncryptedText256 = SecurityMechanisms.AES_Encrypt(
Plaintext,SECRET_KEY_RC4,256);
    end_time = System.nanoTime();
    long AES256_Te = TimeDifference(start_time,end_time)/1000000;
    long AES256_theta = AESEncryptedText256.length();
    start_time = System.nanoTime();
    String AESEncryptedText192 = SecurityMechanisms.AES_Encrypt(
Plaintext,SECRET_KEY_RC4,192);
    end_time = System.nanoTime();
    long AES192_Te = TimeDifference(start_time,end_time)/1000000;
    long AES192_theta = AESEncryptedText192.length();
    start_time = System.nanoTime();
    String AESEncryptedText128 = SecurityMechanisms.AES_Encrypt(
Plaintext,SECRET_KEY_RC4,128);
    end_time = System.nanoTime();
    long AES128_Te = TimeDifference(start_time,end_time)/1000000;
    long AES128_theta = AESEncryptedText128.length();

    //DECRYPTION

    start_time = System.nanoTime();
    String RC4DecryptedText = SecurityMechanisms.RC4Decrypt(
RC4EncryptedText,SECRET_KEY_RC4.getBytes());
    end_time = System.nanoTime();
    long RC4_Td = TimeDifference(start_time,end_time)/1000000;
    start_time = System.nanoTime();
    String AESDecryptedText256 = SecurityMechanisms.AES_Decrypt(
AESEncryptedText256,SECRET_KEY_RC4,256);
    end_time = System.nanoTime();
    long AES256_Td = TimeDifference(start_time,end_time)/1000000;
    start_time = System.nanoTime();
    String AESDecryptedText192 = SecurityMechanisms.AES_Decrypt(
AESEncryptedText192,SECRET_KEY_RC4,192);
    end_time = System.nanoTime();
```

```
        long AES192_Td = TimeDifference(start_time,end_time)/1000000;
        start_time = System.nanoTime();
199     String AESDecryptedText128 = SecurityMechanisms.AES_Decrypt(
AESEncryptedText128,SECRET_KEY_RC4,128);
        end_time = System.nanoTime();
        long AES128_Td = TimeDifference(start_time,end_time)/1000000;
        start_time = System.nanoTime();
        String BFDecryptedText = SecurityMechanisms.BF_decrypt(
BFEncryptedText,SECRET_KEY_RC4);
204     end_time = System.nanoTime();
        long BF_Td = TimeDifference(start_time,end_time)/1000000;
        VerticalSpace();
        VerticalSpace();

209     //Assigning Values to the Vectors
        //ROWS = Encryption Algorithm
        //COLUMNS = Running Instance

        //Encryption Time
214     EncryptionTime[0][x] = AES128_Te;
        EncryptionTime[1][x] = AES192_Te;
        EncryptionTime[2][x] = AES256_Te;
        EncryptionTime[3][x] = RC4_Te;
        EncryptionTime[4][x] = BF_Te;
219     //Decryption Time
        DecryptionTime[0][x] = AES128_Td;
        DecryptionTime[1][x] = AES192_Td;
        DecryptionTime[2][x] = AES256_Td;
        DecryptionTime[3][x] = RC4_Td;
224     DecryptionTime[4][x] = BF_Td;
        //Theta
        theta[0][x] = AES128_theta;
        theta[1][x] = AES192_theta;
        theta[2][x] = AES256_theta;
229     theta[3][x] = RC4_theta;
        theta[4][x] = BF_theta;
    }

    public static String RandomStringGenerator() {
```

```
        int leftLimit = 97; // letter 'a'
        int rightLimit = 122; // letter 'z'
        long targetStringLength = PlaintextLength;
        Random random = new Random();
        String generatedString = random.ints(leftLimit, rightLimit + 1)
                .limit(targetStringLength)
                .collect(StringBuilder::new, StringBuilder::
    appendCodePoint, StringBuilder::append)
                .toString();
        return generatedString;
    }
    public static void VerticalSpace(){
        System.out.println("\n\n");
    }
}
```

Listing B.3: Code for Computing the Cost of Different Security Algorithms

```
package MigrationModel;
import org.bouncycastle.jcajce.provider.digest.Keccak;
import org.bouncycastle.util.encoders.Base64Encoder;
import javax.crypto.*;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.PBEKeySpec;
import javax.crypto.spec.SecretKeySpec;
import java.io.UnsupportedEncodingException;
import java.math.BigInteger;
import java.nio.charset.Charset;
import java.nio.charset.StandardCharsets;
import java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.spec.KeySpec;
import java.util.Base64;

public class SecurityMechanisms {
    //public static String SECRET_KEY;
    public static final String SALT = "ssshhhhhhhhhh!!!!ssshhhhhhhhhhh
    !!!!";
```

```
    //public static int AES_Key_Length;
    public static String AES_cipher_setting;
    public static String SECRET_KEY_BF;
    public static String SECRET_KEY_RC4 = "ThisIsTheRC4SecretKey";
    public static String HASH_ALGO = "SHA-512";
    public static int Keccak_Key_Length = 256;

    private static final String STREAM_ENCRYPTION_ALGORITHM = "ARCFOUR";
    // or "RC4"

    ////////////////////////////////////////      AES
    ////////////////////////////////////////////////

    public static String AES_Encrypt(String strToEncrypt, String
    SECRET_KEY, int AES_Key_Length) {
        try {
            byte[] iv = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
            IvParameterSpec ivspec = new IvParameterSpec(iv);

            SecretKeyFactory factory = SecretKeyFactory.getInstance("
    PBKDF2WithHmacSHA256");
            KeySpec spec = new PBEKeySpec(SECRET_KEY.toCharArray(), SALT.
    getBytes(), 65536, AES_Key_Length);
            SecretKey tmp = factory.generateSecret(spec);
            SecretKeySpec secretKey = new SecretKeySpec(tmp.getEncoded(),
    "AES");

            //Cipher cipher = Cipher.getInstance(AES_cipher_setting);
            Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
            //Cipher cipher = Cipher.getInstance("AES/CBC/NoPadding");
            //Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
            //Cipher cipher = Cipher.getInstance("AES/ECB/NoPadding");
            //Cipher cipher = Cipher.getInstance("AES/CTR/PKCS5Padding");
            //Cipher cipher = Cipher.getInstance("AES/CTR/NoPadding");
            cipher.init(Cipher.ENCRYPT_MODE, secretKey, ivspec);
            //cipher.init(Cipher.ENCRYPT_MODE, secretKey);
            return Base64.getEncoder()
                    .encodeToString(cipher.doFinal(strToEncrypt.getBytes(
    StandardCharsets.UTF_8)));
```

```
        } catch (Exception e) {
            System.out.println("Error while encrypting: " + e.toString());
        }
        return null;
    }

    public static String AES_Decrypt(String strToDecrypt, String
    SECRET_KEY, int AES_Key_Length) {
        try {
            byte[] iv = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
            IvParameterSpec ivspec = new IvParameterSpec(iv);

            SecretKeyFactory factory = SecretKeyFactory.getInstance("
    PBKDF2WithHmacSHA256");
            KeySpec spec = new PBEKeySpec(SECRET_KEY.toCharArray(), SALT.
    getBytes(), 65536, AES_Key_Length);
            SecretKey tmp = factory.generateSecret(spec);
            SecretKeySpec secretKey = new SecretKeySpec(tmp.getEncoded(),
    "AES");

            Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
            //Cipher cipher = Cipher.getInstance("AES/CBC/NoPadding");
            //Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
            //Cipher cipher = Cipher.getInstance("AES/ECB/NoPadding");
            //Cipher cipher = Cipher.getInstance("AES/CTR/PKCS5Padding");
            //Cipher cipher = Cipher.getInstance("AES/CTR/NoPadding");

            cipher.init(Cipher.DECRYPT_MODE, secretKey, ivspec);
            //cipher.init(Cipher.DECRYPT_MODE, secretKey);
            return new String(cipher.doFinal(Base64.getDecoder().decode(
    strToDecrypt)));
        } catch (Exception e) {
            System.out.println("Error while decrypting: " + e.toString());
        }
        return null;
    }

    ////////////////////////////////////       RC4
    ////////////////////////////////////////////////////////////
```

```
    public static byte[] RC4_encrypt(String plaintext) throws
    InvalidKeyException, IllegalBlockSizeException, BadPaddingException,
    NoSuchAlgorithmException, NoSuchPaddingException {
88
        KeyGenerator rc4KeyGenerator = KeyGenerator.getInstance(
    STREAM_ENCRYPTION_ALGORITHM);
        SecretKey secretKey = rc4KeyGenerator.generateKey();
        Cipher rc4 = Cipher.getInstance(STREAM_ENCRYPTION_ALGORITHM);

93      rc4.init(Cipher.ENCRYPT_MODE, secretKey);
        byte[] plaintextBytes = plaintext.getBytes();
        byte[] ciphertextBytes = rc4.doFinal(plaintextBytes);
        //System.out.println("RC4 ciphertext base64 encoded: " + Base64.
    encodeBase64String(ciphertextBytes));
        return ciphertextBytes;
98  }

    public static byte[] RC4_decrypt(byte[] ciphertextBytes) throws
    InvalidKeyException, InvalidAlgorithmParameterException,
    IllegalBlockSizeException, BadPaddingException,
    NoSuchAlgorithmException, NoSuchPaddingException {

        KeyGenerator rc4KeyGenerator = KeyGenerator.getInstance(
    STREAM_ENCRYPTION_ALGORITHM);
103     SecretKey secretKey = rc4KeyGenerator.generateKey();
        Cipher rc4 = Cipher.getInstance(STREAM_ENCRYPTION_ALGORITHM);
        rc4.init(Cipher.DECRYPT_MODE, secretKey, rc4.getParameters());
        byte[] byteDecryptedText = rc4.doFinal(ciphertextBytes);
        return  byteDecryptedText;
108     //String plaintextBack = new String(byteDecryptedText);
        //System.out.println("Decrypted back to: " + plaintextBack);
    }


113 public static String RC4Encrypt(String value, byte[] key) throws
    InvalidKeyException, IllegalBlockSizeException, BadPaddingException,
    NoSuchAlgorithmException, NoSuchPaddingException {

            final Cipher rc4 = Cipher.getInstance("ARCFOUR");
```

```
            rc4.init(Cipher.ENCRYPT_MODE, new SecretKeySpec(key, "ARCFOUR"
    ));
            return Base64.getEncoder().encodeToString(rc4.doFinal(value.
    getBytes()));
118
    }

    public static String RC4Decrypt(String value, byte[] key) throws
    InvalidKeyException, IllegalBlockSizeException, BadPaddingException,
    NoSuchAlgorithmException, NoSuchPaddingException {

123     final Cipher rc4 = Cipher.getInstance("ARCFOUR");
        rc4.init(Cipher.DECRYPT_MODE, new SecretKeySpec(key, "ARCFOUR"));
        return new String(rc4.doFinal(Base64.getDecoder().decode(value)));

    }
128 ////////////////////////////////          SHA HASH
    ////////////////////////////////////////////////////////////////

    public static String Hash (String message) throws
    NoSuchAlgorithmException {
        // getInstance() method is called with algorithm SHA-512
        MessageDigest md = MessageDigest.getInstance(HASH_ALGO);
133
        // digest() method is called
        // to calculate message digest of the input string
        // returned as array of byte
        byte[] messageDigest = md.digest(message.getBytes());
138
        // Convert byte array into signum representation
        BigInteger no = new BigInteger(1, messageDigest);

        // Convert message digest into hex value
143     String hashtext = no.toString(16);

        // Add preceding 0s to make it 32 bit
        while (hashtext.length() < 32) {
            hashtext = "0" + hashtext;
148     }
```

```
        // return the HashText
        return hashtext;
    }

    ////////////////////////////       BlowFish
    ////////////////////////////////////////////////////////////////////
    public static String BF_encrypt(String password, String key) throws
    NoSuchAlgorithmException, NoSuchPaddingException, InvalidKeyException,
     IllegalBlockSizeException, BadPaddingException,
    UnsupportedEncodingException {
        byte[] KeyData = key.getBytes();
        SecretKeySpec KS = new SecretKeySpec(KeyData, "Blowfish");
        Cipher cipher = Cipher.getInstance("Blowfish");
        cipher.init(Cipher.ENCRYPT_MODE, KS);
        String encryptedtext = Base64.getEncoder().encodeToString(cipher.
    doFinal(password.getBytes("UTF-8")));
        return encryptedtext;

    }

    public static String BF_decrypt(String encryptedtext, String key)
    throws NoSuchAlgorithmException, NoSuchPaddingException,
    InvalidKeyException, IllegalBlockSizeException, BadPaddingException {
        byte[] KeyData = key.getBytes();
        SecretKeySpec KS = new SecretKeySpec(KeyData, "Blowfish");
        byte[] ecryptedtexttobytes = Base64.getDecoder().
                decode(encryptedtext);
        Cipher cipher = Cipher.getInstance("Blowfish");
        cipher.init(Cipher.DECRYPT_MODE, KS);
        byte[] decrypted = cipher.doFinal(ecryptedtexttobytes);
        String decryptedString = new String(decrypted, Charset.forName("
    UTF-8"));
        return decryptedString;

    }

    ////////////////////      BOUNCY CASTLE
```

```
    ////////////////////////////////
    public static String hashKeccak(String data) {
        byte[] dataBytes = data.getBytes();
        Keccak.DigestKeccak md = new Keccak.DigestKeccak(Keccak_Key_Length
    );
        md.reset();
        md.update(dataBytes, 0, dataBytes.length);
        byte[] hashedBytes = md.digest();
        BigInteger no = new BigInteger(1, hashedBytes);
        String hashtext = no.toString(16);
        return hashtext;
    }
}
```

Listing B.4: Class with all the Cryptographic Algorithms and Security Mechanisms

B.4 Simulating Migration Time Variation for Different Security Settings/ Algorithms using MatLab

The Listing B.5 depicts the MatLab code for running a simulation on different security settings or algorithms for plotting their migration time variation against different critical BW utilizations. The results of this simulation can be seen in Fig. 6.8.

```
Channel_Capacity = 1*(10^10); %bps
s = (4.17)*(10^9); %bytes
s_bits = 8*(16)*(10^9); %bits
USA_Size = 9833516; %km^2
USA_map_Size = 41*(10^12); %TB

%Channel Critical BW Utilization Values
BW = 0.7:0.02:1.0;

BW_length = length(BW);
```

```
%States
S = [0 1 2 3 4];

%Markov States
S_BW = [0 1 2 3 4; 0.867 0.9 0.933 0.967 1];

%Transition Probability Matrix
TPM = [0.0484377 0.045735199 0.044003627 0.043398172 0.044140315];

disp(TPM(3));

%Assuming a Critical Scenario where the Occupied Channel BW >= 0.867 %

I = s_bits;
Padding_Scheme = 0;

SO_AES = AES_SO(I,0);

disp('The Input Size : ');
disp(I);
disp('The Output Size : ');
disp(SO_AES);

%Assuming Stream Ciphers has a null overhead
SO_SC = I;
T_SC = 200*2;

%Variation for Plaintext
T_M_Plaintext = MigrationTime(Channel_Capacity,BW,s_bits);



%Variation for AES
SO_AES = AES_SO(s_bits,0);
ET_AES = T_AES(s_bits);

T_M_AES = MigrationTime(Channel_Capacity,BW,SO_AES);

%Variation for AES Padding
SO_AES_PADDING = AES_SO(s_bits,1);
```

```
T_M_AES_PADDING = MigrationTime(Channel_Capacity,BW,SO_AES_PADDING);

%Variation for BF
SO_BF = BF_SO(s_bits);

T_M_BF = MigrationTime(Channel_Capacity,BW,SO_BF);

%Variation for RC4
SO_RC4 = RC4_SO(s_bits);

T_M_RC4 = MigrationTime(Channel_Capacity,BW,SO_RC4);


%Variation for high
SO_High = High_SO(s_bits);

T_M_High = MigrationTime(Channel_Capacity,BW,SO_High);

%Scheme_1
T_M_Scheme_1 = MigrationTime_scheme_1(Channel_Capacity,BW,s_bits);

%Scheme_1
T_M_Scheme_2 = MigrationTime_scheme_2(Channel_Capacity,BW,s_bits);

%Plotting
plot(BW, T_M_Plaintext);
hold on
plot(BW, T_M_AES);
hold on
plot(BW, T_M_AES_PADDING);
hold on
plot(BW, T_M_BF);
hold on
plot(BW, T_M_RC4);
hold on
plot(BW, T_M_High);
hold on
plot(BW, T_M_Scheme_1);
```

```
hold on
plot(BW, T_M_Scheme_2);
hold off

function T_M = MigrationTime_scheme_1(ChannelCapacity,BandwidthUtilization
    ,MigratingFileSize)
x = length(BandwidthUtilization);
T_M = 1:1:x;
j=1;
    while (j < (x+1))
        if BandwidthUtilization(j) < 0.75
            MigrationFileSize = High_SO(MigratingFileSize);
            T_M(j) = MigrationFileSize/(ChannelCapacity*(1-
    BandwidthUtilization(j)));
        elseif BandwidthUtilization(j) < 0.9
           MigrationFileSize = AES_SO(MigratingFileSize,0);
           T_M(j) = MigrationFileSize/(ChannelCapacity*(1-
    BandwidthUtilization(j)));
        else
           MigrationFileSize = RC4_SO(MigratingFileSize);
           T_M(j) = MigrationFileSize/(ChannelCapacity*(1-
    BandwidthUtilization(j)));
        end
        j=j+1;
    end
end

function T_M = MigrationTime_scheme_2(ChannelCapacity,BandwidthUtilization
    ,MigratingFileSize)

x = length(BandwidthUtilization);
T_M = 1:1:x;

j=1;
    while (j < (x+1))
        if BandwidthUtilization(j) < 0.75
            MigrationFileSize = AES_SO(MigratingFileSize,0);
            T_M(j) = MigrationFileSize/(ChannelCapacity*(1-
    BandwidthUtilization(j)));
```

```
        elseif BandwidthUtilization(j) < 0.9
           MigrationFileSize = AES_SO(MigratingFileSize,0);
           T_M(j) = MigrationFileSize/(ChannelCapacity*(1-
    BandwidthUtilization(j)));
        else
           MigrationFileSize = RC4_SO(MigratingFileSize);
           T_M(j) = MigrationFileSize/(ChannelCapacity*(1-
    BandwidthUtilization(j)));
        end
        j=j+1;
    end
end
function T_M = MigrationTime(ChannelCapacity,BandwidthUtilization,
    MigratingFileSize)

T_M = MigratingFileSize./(ChannelCapacity*(1-BandwidthUtilization));

end

function O = AES_SO(I,Padding_Scheme)

if (Padding_Scheme == 0)

O_min = 24;
delta = 20;
divisor = 48;

O_max = (O_min + 2*delta);

    if (I < 48)
        I_hat = 0;
        I_dash = I;
    else
        I_hat = fix(I/divisor);
        I_dash = rem(I,divisor);
    end
        O = O_max * I_hat;
    if (I_dash < 16)
        O = O + O_min;
```

```
    elseif (I_dash < 32)
        O = O + O_min + delta;
    else
        O = O + O_min + 2 * delta;
    end
else

O_min = 24;
delta = 20;
divisor = 48;

    if (rem(I,16) == 0)
        O_max = (O_min + 2*delta);
        if (I < 64)
            I_hat = 0;
            I_dash = I;
        else
            I_hat = fix(I/divisor);
            I_dash = rem(I,divisor);
        end
O = O_max * I_hat;
        if (I_dash == 16)
            O = O + O_min;
        elseif (I_dash == 32)
            O = O + O_min + delta;
        else
            O = O + O_min + 2 * delta;
        end
    end
end
end


function O = BF_SO(I)
O_min = 12;
delta = 8;
divisor = 24;

O_max = (2*O_min + delta);
```

```
if (I < 24)
    I_hat = 0;
    I_dash = I;
else
    I_hat = fix(I/divisor);
    I_dash = rem(I,divisor);
end
O = O_max * I_hat;

if (I_dash < 8)
    O = O + O_min;
elseif (I_dash < 16)
    O = O + 2 * O_min;
else
    O = O + 2 * O_min + delta;
end
end

function O = RC4_SO(I)
O = I*1.005;
end

function O = High_SO(I)
O = I*1.5;
end

function T = T_AES(I)
    if (I > (10^9))
        I_hat = fix(I/(10^9));
        I_dash = rem(I,(10^9));
        T = I_hat*5200 + ((5200-194)/(100))*I_dash;
    else
        T = ((5200-194)/(100))*I;
    end
end
```

Listing B.5: MatLAB code for migration time variation simulation of different SSs

B.5 Prototype MEC SMSF Implementation

For validating the proposed security management model, a prototype framework was implemented. The programs related to this framework are presented in the GitHub link: `https://github.com/Pasika87/MEC-SMSF.git`. Fig. B.1 shows the respective ER diagram of the program.

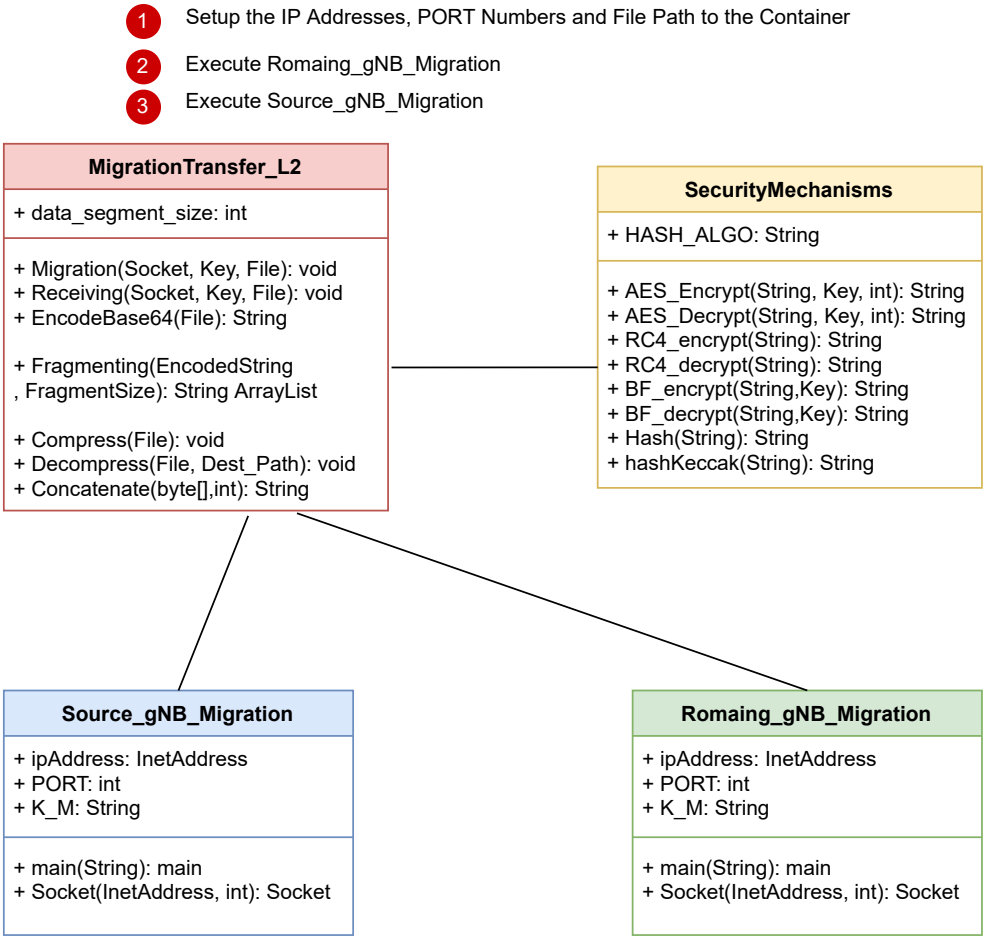


Figure B.1: MEC-SMSF Migration ER Relation and Execution Instructions

Fig B.2, Fig B.3, Fig B.4, Fig B.5, and Fig B.6 shows the respective CLI outcomes of the developed program that indicate the statistics and steps of the migration process from gNB_S and gNB_R perspectives.

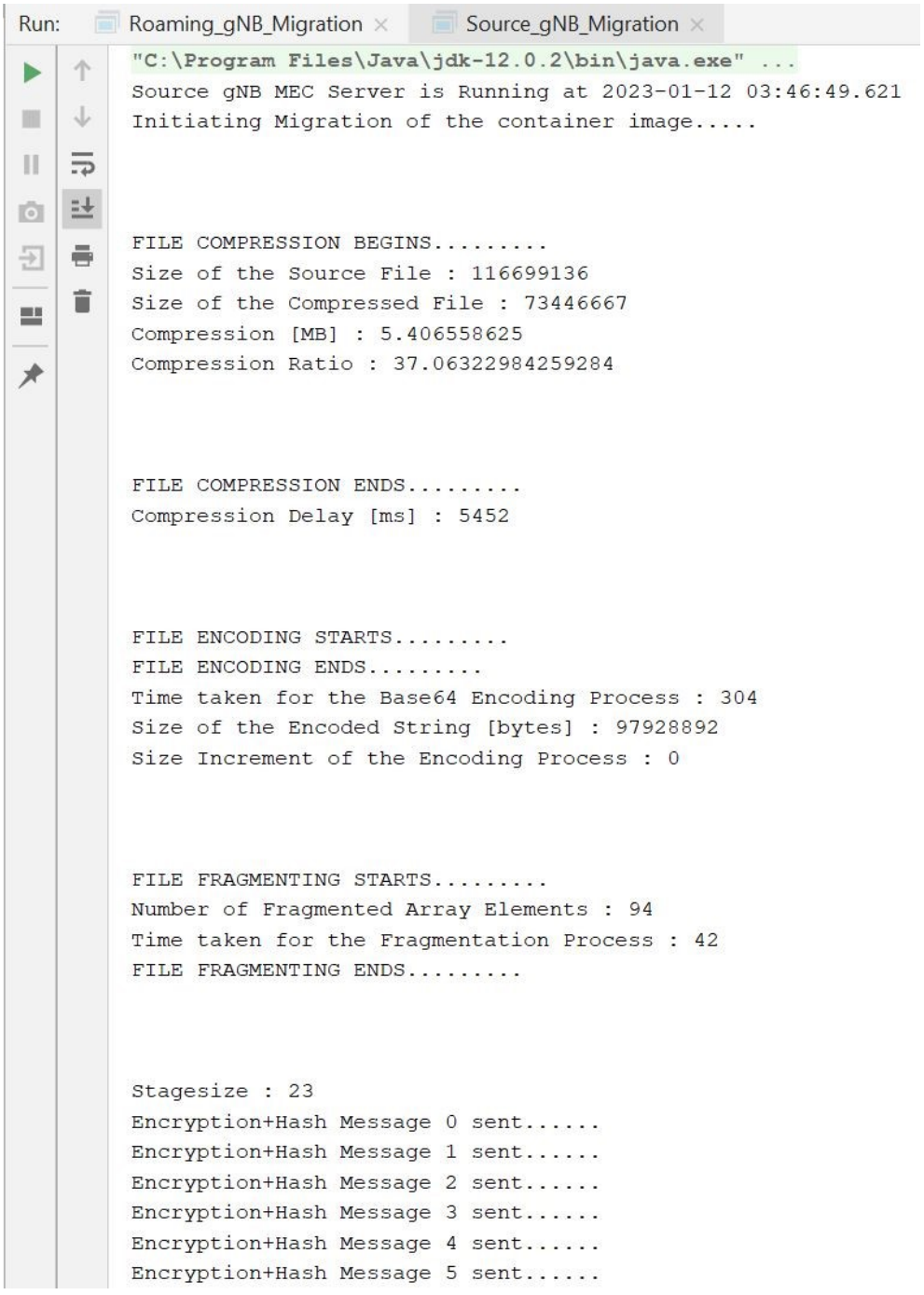


Figure B.2: Command Line Interface Outcome 1 of the gNB_S Migrating

```
Encryption+Hash Message "+i+" sent......
..............SECURITY PROFILE SWITCHED.............
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
Encryption+Hash Message "+i+" sent......
FILE MIGRATION ENDS.........
Migration Delay [ms] : 6109



Time taken for migration - sending Processing [ms]: 11910
File Send Start Time : 0
File Send End Time : 0

Process finished with exit code 0
```

Figure B.3: Command Line Interface Outcome 2 of the gNB_S Migrating

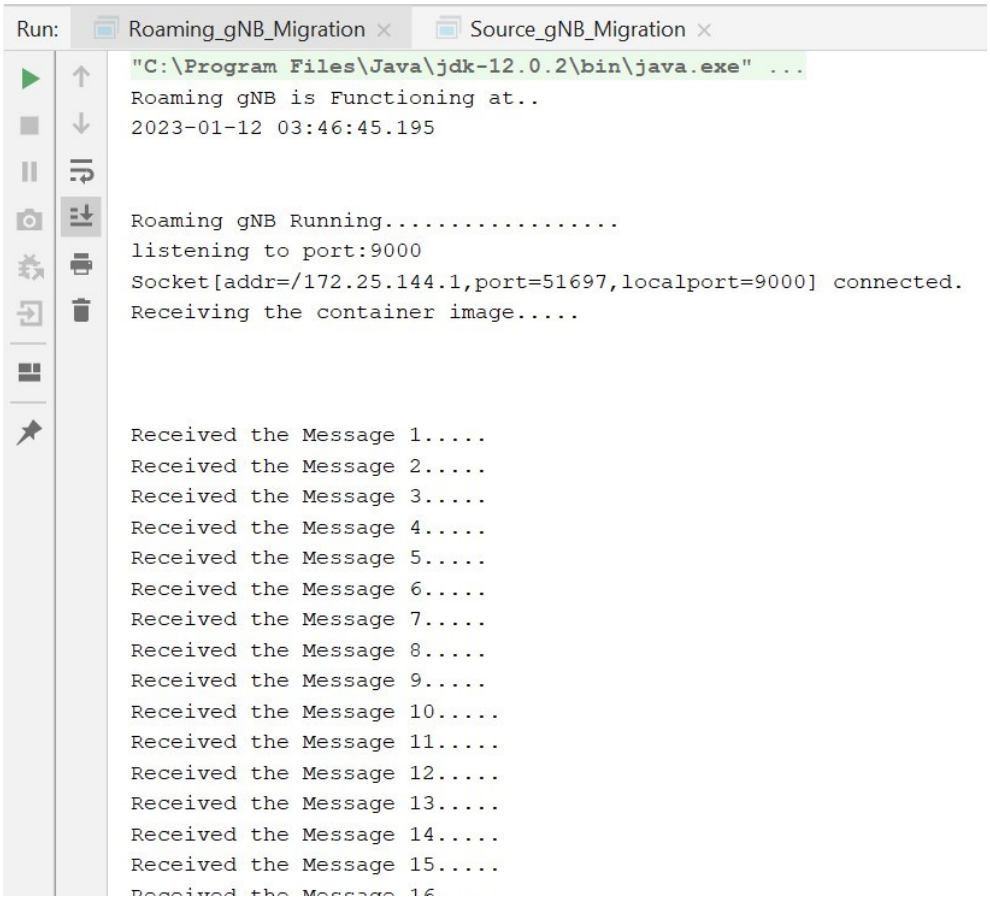


Figure B.4: Command Line Interface Outcome 1 of the gNB_R Migrating

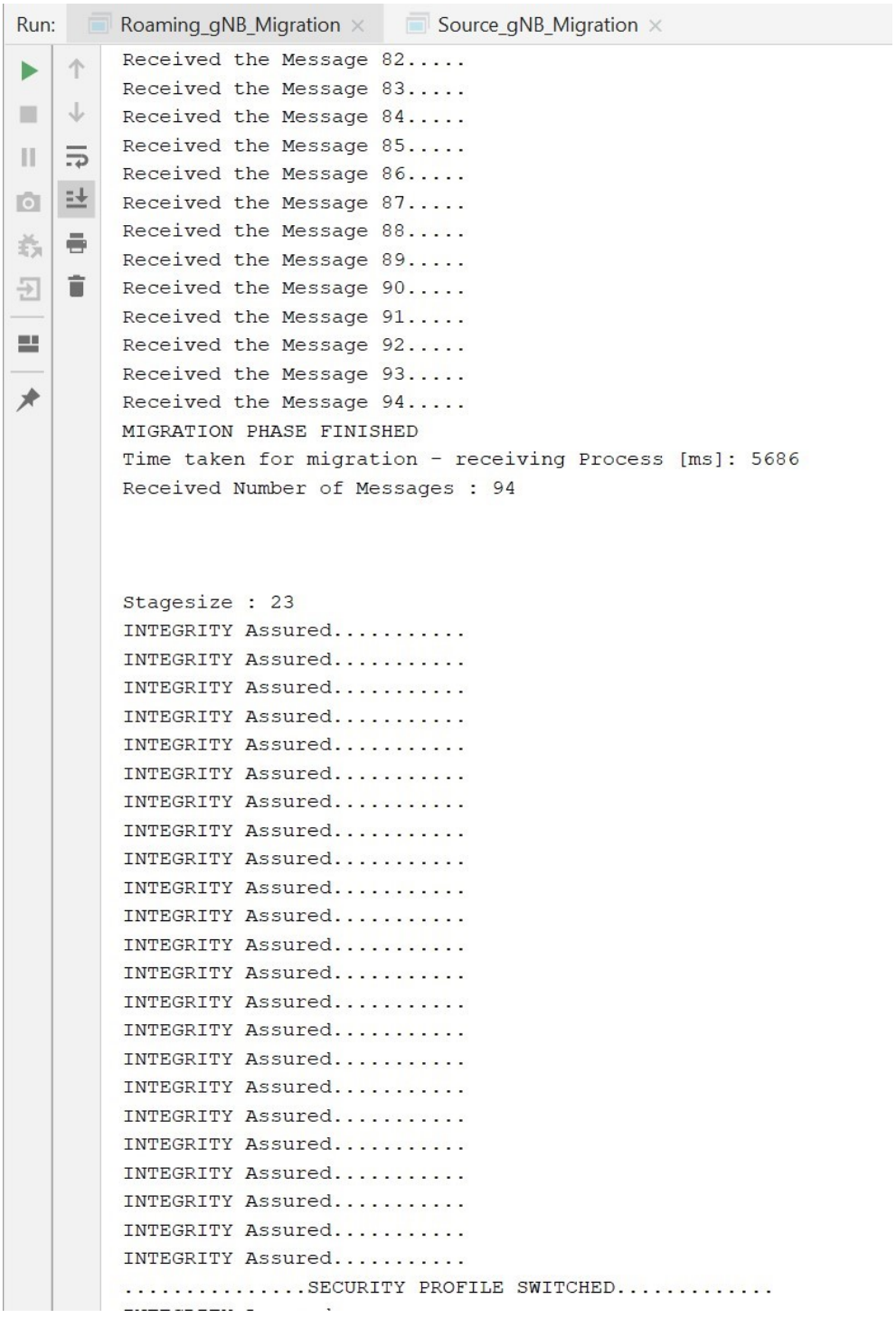


Figure B.5: Command Line Interface Outcome 2 of the gNB_R Migrating

```
..............SECURITY PROFILE SWITCHED.............
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
INTEGRITY Assured..........
DECRYPTION COMPLETE......
File Concatination/ Formation COMPLETE...........
Time taken for Decryption [ms]: 8237



FILE DECODING BEGINS.........
FILE DECODING ENDS.........
Time taken for Base64 Decoding [ms] : 764



FILE DECOMPRESSION BEGINS.........
FILE DECOMPRESSION ENDS.........
Time taken for Decompression [ms] : 1314



Process Time [ms]: 22230
File Receive Start Time : 0
File Receive End Time : 0

Process finished with exit code 0
```


Figure B.6: Command Line Interface Outcome 3 of the gNB_R Migrating