David José
Araújo Ferreira

Integração de Dispositivos *Wi-Fi-Only* em Redes 5G: Abordagem aos Desafios de Autenticação e Gestão de Identidade

Integration of Wi-Fi-Only Devices in 5G Core Networks: Addressing Authentication and Identity Management Challenges

Universidade de Aveiro
2025

David José Araújo Ferreira

Integração de Dispositivos *Wi-Fi-Only* em Redes 5G: Abordagem aos Desafios de Autenticação e Gestão de Identidade

Integration of Wi-Fi-Only Devices in 5G Core Networks: Addressing Authentication and Identity Management Challenges

Dissertation presented to the University of Aveiro in fulfillment of the necessary requirements for the obtaining of the degree of Master in Cybersecurity, conducted under the scientific supervision of Doctor Daniel Nunes Corujo, Associate Professor with Habilitation of the Departamento de Eletrónica, Telecomunicações e Informática at the University of Aveiro,
Doctor Francisco Fontes, Senior Consultant from Altice Labs .

o júri / **the jury**

presidente / president

Prof. Doutor João Antunes da Silva
professor associado da Universidade de Aveiro

vogais / examiners committee

Prof. Doutor João Antunes da Silva
professor associado da Universidade de Aveiro

Prof. Doutor João Antunes da Silva
professor associado da Universidade de Aveiro

Prof. Doutor João Antunes da Silva
professor associado da Universidade de Aveiro

Prof. Doutor João Antunes da Silva
professor associado da Universidade de Aveiro

Prof. Doutor João Antunes da Silva
professor associado da Universidade de Aveiro

agradecimentos /
acknowledgements

Em primeiro lugar, quero expressar a minha profunda gratidão ao meu orientador, Prof. Doutor Daniel Nunes Corujo, por ter acreditado em mim e no meu trabalho desde o primeiro momento. Desde os tempos da licenciatura, onde também me orientou no projeto final, tem sido um exemplo de confiança, orientação e incentivo constante. O seu acompanhamento próximo ao longo destes mais de dois anos foi fundamental para a concretização deste trabalho, e estarei sempre imensamente grato pelas oportunidades que me proporcionou.

Agradeço igualmente ao Doutor Francisco Fontes, pelo apoio contínuo ao longo deste projeto e pela disponibilidade incansável em mobilizar os recursos da Altice Labs, permitindo que esta investigação atingisse os seus objetivos. A sua confiança e dedicação criaram um ambiente ideal que em muito contribuiu para os resultados alcançados.

Ao Miguel Freitas, da Altice Labs, deixo um agradecimento muito especial. Um colega e engenheiro cuja competência técnica, dedicação e capacidade de execução continuo a admirar. A sua colaboração foi essencial para o desenvolvimento dos projetos em que trabalhámos juntos, e aprendi imenso com ele ao longo destes dois anos.

Por fim, e talvez mais importante, quero agradecer à minha família. À minha mãe, Ana Araújo, e à minha irmã, Ana Catarina, pelo apoio incondicional que sempre me deram, não só ao longo deste percurso académico, mas em toda a minha vida. Este trabalho também é vosso, assim como o futuro que me esforço para contruir.

A todos, muito obrigado.

palavras-chave

Rede 5G, Integração Wi-Fi, Non-3GPP Access, Autenticação, Gestão de Identidade, EAP-TLS, RADIUS, PDU Sessions, 5G Residential Gateway, Identidade Proxy, Open5GS, UERANSIM

resumo

À medida que as redes de quinta geração (5G) evoluem, a integração de dispositivos exclusivamente Wi-Fi, sem capacidades como um *Universal Subscriber Identity Module* (USIM), no núcleo da rede 5G (5GC) continua a representar um desafio significativo, especialmente em ambientes empresariais e residenciais. Esta dissertação aborda a lacuna existente nas normas atuais do 3GPP, propondo uma estrutura inovadora que permite a autenticação segura e a gestão de identidade para este tipo de dispositivos. A solução desenvolvida baseia-se em métodos do *Extensible Authentication Protocol* (EAP), com ênfase no EAP-TLS, e num mecanismo de identidade por *proxy* através de sessões de *Protocol Data Unit* (PDU), permitindo a identificação consistente dos dispositivos e a gestão do tráfego dentro da 5GC. A arquitetura foi implementada utilizando componentes virtualizados de redes 5G e validada através de cenários experimentais que simularam o registo, mapeamento de sessões e gestão do ciclo de vida de dispositivos não autenticáveis de acesso não-3GPP (NAUN3). Os resultados demonstram que a estrutura proposta suporta eficazmente a integração segura, o isolamento de tráfego e o controlo dinâmico de sessões, oferecendo um caminho viável para a conectividade segura e inclusiva de dispositivos *Wi-Fi-only* no ecossistema 5G.

keywords

5G Network, Wi-Fi Integration, Non-3GPP Access, Authentication, Identity Management, EAP-TLS, RADIUS, PDU Sessions, 5G Residential Gateway, Proxy Identity, Open5GS, UERANSIM

abstract

As fifth-generation (5G) networks continue to evolve, integrating Wi-Fi, only devices, lacking Universal Subscriber Identity Module (USIM) capabilities, into the 5G Core (5GC) remains a significant challenge, particularly in enterprise and residential environments. This dissertation addresses the gap in existing 3GPP standards by proposing a novel framework that enables secure authentication and identity management for such devices. The proposed solution leverages Extensible Authentication Protocol (EAP) methods, particularly EAP-TLS, alongside a proxy identity mechanism using Protocol Data Unit (PDU) sessions, allowing for consistent device identification and traffic management within the 5GC. The architecture was implemented using virtualized 5G components and validated through a series of experiments simulating onboarding, session mapping, and lifecycle management of non-authenticable non-3GPP (NAUN3) devices. Results demonstrate that the framework effectively supports secure integration, traffic isolation, and dynamic session control, offering a practical path toward inclusive and secure 5G connectivity for legacy Wi-Fi devices.

acknowledgement of use of AI tools

Recognition of the use of generative Artificial Intelligence technologies and tools, software and other support tools.

I acknowledge the use of Perplexity AI (Perplexity AI, Inc., https://www.perplexity.ai/) and Gemini (https://gemini.google.com/) to summarize the initial notes, document structuring, spell and phrashing check and to proofread the final draft.

Contents

List of Figures

List of Tables

List of Code Blocks

Glossary

2G second-generation
3G third-generation
3GPP 3rd Generation Partnership Project
4G fourth-generation
5G fifth-generation
5G-AKA 5G - Authentication and Key Management
5G-GUTI 5G Globally Unique Temporary Identifier
5G-RG 5G Residential Gateway
5G-S-TMSI 5G Short Temporary Mobile Subscriber Identity
5G-TMSI 5G Temporary Mobile Subscriber Identity
5GC 5G Core Network
AF Application Function
AKA Authentication and Key Agreement
AMF Access and Mobility Management Function
AN Access Network
API Application Programming Interface
AP Access Point
APN Access Point Name
ARP Address Resolution Protocol
ARPF Authentication Credential Repository and Processing Function
AT Attention
AUSF Authentication Server Function
AV Authentication Vector
CA Certification Authority
CGID Connectivity Groups Identifier
CLI Command Line Interface
CN Core Network
CP Control Plane
CPU Central Processing Unit
CRG Customer Residential Gateway
DHCP Dynamic Host Configuration Protocol
DN Data Networks
DNN Data Network Name
DSL Digital Subscriber Line
DTLS Datagram Transport Layer Security
EAP Extensible Authentication Protocol
EAP-5G Extensible Authentication Protocol 5G
EAP-AKA' Extensible Authentication Protocol - Authentication and Key Management
EAP-TLS Extensible Authentication Protocol - Transport Layer Security
EAP-TTLS Extensible Authentication Protocol - Tunneled Transport Layer Security
EAPOL Extensible Authentication Protocol Over LAN
ECIES Elliptic Curve Integrated Encryption Scheme
EMSK Extended Master Session Key
eBPF Extended Berkeley Packet Filter
eMBB enhanced Mobile Broadband
eNodeB evolved Node B
EPC Evolved Packet Core
EPS Evolved Packet System
EPS-AKA Evolved Packet System - Authentication and Key Management
EUI-64 Extended Unique Identifier-64
FN-CRG Fixed Network Customer Residential Gateway
FN-RG Fixed Network Residential Gateway
FWA Fixed Wireless Access
GB Gigabyte
GCI Global Cable Identifier
GLI Global Line Identifier
gNB Next Generation Node B
GTP General Packet Radio Service Tunneling Protocol
GTP-U General Packet Radio Service Tunnelling Protocol - User Plane
GUAMI Globally Unique AMF Identifier
GUTI Globally Unique Temporary Identity
HN Home Network
HPLMN Home Operator Public Land Mobile Network
HSS Home Subscriber Server
IEEE Institute of Electrical and Electronics Engineers
IKE Internet Key Exchange
IKEv2 Internet Key Exchange version 2
IMEI International Mobile Equipment Identity
IMSI International Mobile Subscriber Identity

IoT Internet of Things
IP Internet Protocol
IPSec IP security
IPSec SA IPSec Security Association
IPSec SA (NWt) IPSec Security Association Network Termination
IPv4 Internet Protocol Version 4
ISP Internet Service Provider
KPI Key Performance Indicator
LAN Local Area Network
LTE Long-Term Evolution
MAC Medium Access Control
MCC Mobile Country Code
ME Mobile Equipment
MME Mobility Management Entity
mMTC massive machine-type communications
MNC Mobile Network Code
MSIN Mobile Subscriber Identification Number
N3IWF Non-3GPP Interworking Function
N5CW Non-5G Capable over WLAN
N5GC Non-5G Capable
NAI Network Access Identifier
NAS Non-Access Stratum
NAT Network Address Translation
NAUN3 Non-Authenticable Non-3GPP
NDS/IP Network Domain Security/IP
NEF Network Exposure Function
NF Network Function
NFV Network Functions Virtualization
NG-RAN Next Generation Radio Access Networks
NGAP Next Generation Access and Service Management Protocol
NR New Radio
NRF NF Repository Function
NSI Network Specific Identifier
NSSF Network Slice Selection Function
OPC Operator Key
PCC Policy and Charging Control
PCF Policy Control Function
PDP Packet Data Protocol
PDU Protocol Data Unit
PEI Permanent Equipment Identifier
PIN Personal Identification Number
PKCS12 Public-Key Cryptography Standards 12
PKI Public Key Infrastructure
PLMN Public Land Mobile Network
PSK Pre-Shared Key
QMAP Qualcomm Multiplexing and Aggregation Protocol
QMI Qualcomm MSM Interface
QoS Quality of Service
RADIUS Remote Authentication Dial-In User Service
RAM Random Access Memory
RAN Radio Access Network
RG Residential Gateway
S-NSSAI Single Network Slice Selection Assistance Information
SBA Service Based Architecture
SBC Single Board Computer
SDN Software Defined Network
SEAF Security Anchor Function
SIDF Subscription Identifier De-concealing Function
SMF Session Management Function
SN Serving Network
SNAT Source Network Address Translation
SNPN Service Network Public Land Mobile Network
SSH Secure Shell
SSID Service Set Identifier
SST Service Selection Tunneling
SUCI Subscriber Concealed Identifier
SUPI Subscription Permanent Identifier
TCP Transmission Control Protocol
TLS Transport Layer Security
TNAN Trusted Non-3GPP Access Network
TNAP Trusted Non-Access Stratum Proxy
TNGF Trusted Non-3GPP Gateway Function
TSN Time-Sensitive Networking
TWIF Trusted WLAN Interworking Function
UDM Unified Data Management
UDR Unified Data Repository
UE User Equipment
UICC Universal Integrated Circuit Card
UDP User Datagram Protocol
UP User Plane
UPF User Plane Function
uRLLC ultra-reliable low-latency communications
URSP User Registration Service Protocol
USIM Universal Subscriber Identity Module
USB Universal Serial Bus
UTC Universal Time Code
VLAN Virtual Local Area Network
VM Virtual Machine
W-5GAN Wireline 5G Access Network
W-AGF Wireline Access Gateway Function
W-CP Wireline Control Plane
WBA Wireless Broadband Alliance
WLAN Wireless Local Area Network
WPA Wi-Fi Protected Access

CHAPTER 1

Introduction

This chapter focuses on contextualizing the current challenges of integrating Wi-Fi-only devices in fifth-generation (5G) networks and the envisioned solution to address these limitations via a novel solution that aims to bridge the gap between these legacy devices and the existing authentication mechanisms implemented in 5G.

1.1 Background and Context

In recent years, 5G wireless technology as shown to have the potential to revolutionize telecommunications. It offers higher bandwidth, faster speeds, and lower delays, supporting areas such as enhanced Mobile Broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable low-latency communications (uRLLC).

Additionally, it is transforming private networks, which have traditionally relied on legacy wired or wireless Ethernet [1]. Features like tighter security, higher reliability, and Time-Sensitive Networking (TSN) are crucial to meeting Industry 4.0 requirements for wireless connectivity. However, how can the industry bridge the gap between existing Non-5G Capable (N5GC) devices, which current networks rely on, and new 5G Core Networks (5GCs)?

5G is not only a revolution in radio network infrastructure, but also in the core network. Now based on Service Based Architecture, using Network Functions Virtualization (NFV) and Software Defined Network (SDN), it is tailored to minimize cost and maximize utilization and elasticity of the infrastructure by separating the User Plane (UP) functions from the Control Plane (CP) functions. This architecture supports various access nodes such as native New Radio (NR), Long-Term Evolution (LTE) accesses, and non-3rd Generation Partnership Project (3GPP) interworking functions that facilitate connectivity from untrusted Wireless Local Area Networks (WLANs). [2]

As wireless networks evolve, the convergence of 5G with existing Wi-Fi infrastructures becomes increasingly critical. However, current standards established by the 3GPP do not adequately address the integration of Wi-Fi-only devices, that lack Universal Subscriber Identity Module (USIM) capabilities, into the 5GC network. [3] This limitation is particularly

evident in enterprise environments where many devices operate solely on Wi-Fi. To fully realize the potential of the technology, it is essential to develop solutions that enable integration of Wi-Fi-only devices into the 5G ecosystem. This includes addressing challenges related to authentication mechanisms, device identity, and overall interoperability between different network types.

1.2 Problem Statement

The current 3GPP standards [4] lack the ability for integrating Wi-Fi-only devices without USIM into the 5GC, creating a significant gap in connectivity. This limitation is problematic in enterprise environments, where many devices operate only on Wi-Fi and do not possess USIM capabilities. The Wireless Broadband Alliance (WBA) has identified this issue [3], recommending that 3GPP should develop procedures to support Wi-Fi-only User Equipment (UE) using non-International Mobile Subscriber Identity (IMSI) based identity and authentication methods such as Extensible Authentication Protocol - Transport Layer Security (EAP-TLS) or Extensible Authentication Protocol - Tunneled Transport Layer Security (EAP-TTLS). Addressing this challenge is crucial for enabling integration of diverse device types into the 5G ecosystem.

1.3 Research Objectives

The main goal of this dissertation is to explore and develop solutions for integrating Wi-Fi-only devices without USIM into the 5GC infrastructure. To achieve this, the following objectives have been identified:

1. Investigate authentication mechanisms compatible with both 5G and Wi-Fi networks:
    - Analyze existing authentication methods such as EAP-TLS and EAP-TTLS for their applicability in a converged 5G-Wi-Fi environment.
    - Research the current and ongoing development on 5G and Wi-Fi authentication interoperability standards.
    - Explore potential modifications or extensions to these methods to ensure seamless authentication across different network types.
2. Develop a method for managing device identity that works across 5G and non-3GPP networks:
    - Investigate the possibility of designing an extended Network Access Identifier (NAI) or alternative identifier that can accommodate Wi-Fi-only devices while maintaining compatibility with the 5G infrastructure.
    - Investigate the possibility of generating a pseudo-Subscriber Concealed Identifier (SUCI) and pseudo-Subscription Permanent Identifier (SUPI) for N5GC devices that follows the NAI format and serves a similar function in the authentication flow.
    - Investigate the possibility of using existing mechanisms that can replace 5G-specific unique identifiers.

3. Propose extensions or alternatives to existing protocols:
    - Investigate the possibility for mapping existing N5GC device identifiers (e.g., Medium Access Control (MAC) addresses) to a format compatible with the 5G authentication framework.
    - Explore the potential for enhancing or creating new Extensible Authentication Protocol (EAP) methods specifically designed for N5GC devices in a 5G context.

1.4 Dissertation Structure

This document explores the challenge of integrating Wi-Fi-only devices into the 5GC. We begin by examining the current landscape of 5G and Wi-Fi integration, focusing on authentication mechanisms and their limitations. Building on this foundation, a framework to address these challenges is proposed, detailing the targeted approach for security. The solution is then put to the test, presenting experimental results and comparing them with existing methods. Finally, a reflection on achieved contributions is given, acknowledging the boundaries of the work, and suggest possible avenues for future research. Through this journey, the aim is to meaningfully contribute to the ongoing convergence of 5G and Wi-Fi technologies.

CHAPTER 2

State of the Art

This chapter will provide a comprehensive review of current 5G and Wi-Fi integration efforts, existing authentication mechanisms, and challenges in device identification, authentication and authorization. It will also explore recent developments and proposed solutions in the field, setting the context for our research.

2.1 Why 4G needed improved security?

From the point of view of authentication, a cellular network consists of three main components (see Figure 2.1): UE, a Serving Network (SN), and a Home Network (HN).

The UE refers to devices like smartphones, tablets, or IoT devices equipped with a Universal Integrated Circuit Card (UICC) hosting at least a USIM storing security related artifacts, including a cryptographic key that is shared with the subscriber's home network. These devices connect to the network over radio signals. In fourth-generation (4G) LTE networks, these signals utilize specific frequency bands allocated for 4G communication.

The SN includes network components that establish communication and provide services to the UE in a specific geographic area. Key elements of the SN are the evolved Node B (eNodeB) and the Mobility Management Entity (MME).

- The eNodeB is a base station that manages the radio connection between the UE and the network. It handles tasks like scheduling radio resources, modulating and demodulating signals, and ensuring reliable data transmission over the air interface.
- The MME is a core network element responsible for managing signaling between the UE and the core network. It plays a key role in tasks such as authenticating the user, establishing bearers (data pathways), and ensuring mobility by managing handovers between eNodeBs as the UE moves.

The HN refers to the network operated by the user's mobile service provider (e.g., MEO, Vodafone, or NOS). Among other functions, it stores subscriber information in a database called the Home Subscriber Server (HSS).

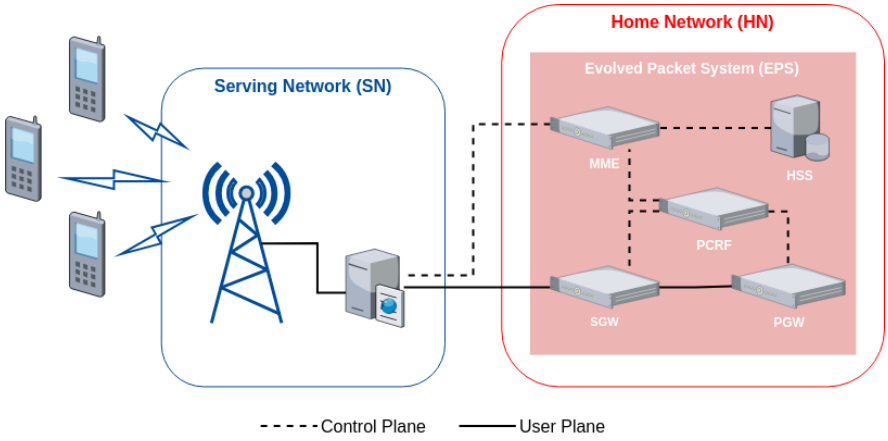


Figure 2.1: 4G Cellular Network Architecture

- The HSS is a critical component that contains subscription-specific data, such as subscription profiles, service entitlements, and cryptographic keys. These keys are used during the authentication process to verify and authorize, under specific conditions, access to that user in the network. The HSS communicates with the SN to authenticate the UE using protocols like Diameter over an IP-based system. This ensures secure and efficient exchange of authentication and session-related information.

Together, these core components form the Evolved Packet System (EPS) [5], the architecture underlying 4G LTE networks. The EPS enables seamless connectivity and service delivery by integrating the radio access network (eNodeBs) with the core network components (e.g., MME and HSS). This design ensures that authentication, data management, and mobility are handled efficiently while providing high-speed, low-latency connections for the UE.

Communication between the SN and HN over the IP network is facilitated by core network protocols. The SN sends a request to the HSS containing the UE's credentials (e.g., IMSI). The HSS uses its stored keys to generate authentication vectors, which are then sent back to the SN. The SN uses these vectors to authenticate the UE and establish a secure connection.

Prior generations to 4G, especially in Radio Access Networks (RANs), have faced significant security and privacy challenges. One major issue was the lack of network authentication in second-generation (2G) [6], which allowed attackers to perform network spoofing using fake base stations. For example, a fake base station could advertise a stronger signal and lure UE away from its legitimate network, enabling the attacker to send fraudulent text messages to the user.

Another issue was the lack of integrity protection for signaling messages [6], which left them vulnerable to spoofing and tampering. For instance, fake base stations could send unprotected Identity Request messages (a Non-Access Stratum (NAS) signaling message in LTE) to steal permanent UE identifiers, such as the IMSI.

Additionally, certain messages lacked confidentiality [6], resulting in privacy violations. For example, unencrypted paging messages could be intercepted to detect a user's presence and track their precise location.

To mitigate these vulnerabilities, the 3GPP introduced the Authentication and Key Agreement (AKA) protocol, which ensures entity authentication, message integrity, and message confidentiality. AKA employs a challenge-response mechanism based on a symmetric key shared between the subscriber and their home network which is pre stored at the UE in the UICC/USIM and in the network at the HSS. It also derives cryptographic keying materials to protect both signaling messages and user plane data, including communications over radio channels. This protocol significantly enhances security and privacy in mobile networks.

In 4G Evolved Packet System - Authentication and Key Management (EPS-AKA), despite the enhancements brought by the 3GPP AKA protocol, two significant flaws remain. First, during the initial stage of the authentication process (the flow is shown in Figure 2.2), the UE must transmit its identity, specifically its IMSI, to the serving network. This identity is sent over the radio network without encryption, leaving it vulnerable to interception [5]. Although the use of a temporary identifier, such as the Globally Unique Temporary Identity (GUTI), is intended to mitigate this risk [7], researchers have demonstrated that GUTI allocation is flawed in that the identifiers either do not change frequently enough [8] or are assigned in predictable patterns [9].

Second, during the authentication decision, the home network may provide an Authentication Vector (AV), but this value is not directly included in the decision-making process, which is handled solely by the serving network [10].

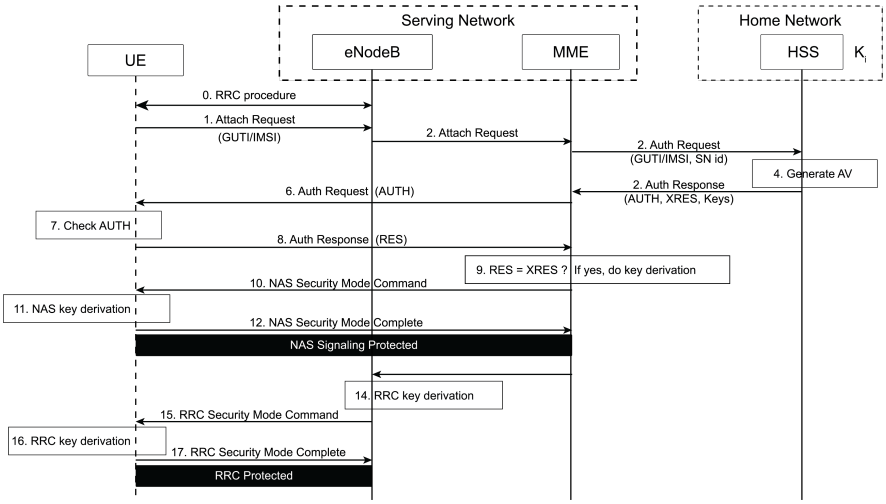


Figure 2.2: 4G Authentication Procedure

2.2 5G Architecture and Security Framework

The 5G System architecture, presented in Figure 2.3, is designed to exploit advanced techniques such as NFV and SDN. Besides, it separates CP and UP functions, enabling independent scalability, evolution, and flexible deployments in centralized or distributed locations. The architecture adopts a modular function design to support efficient network slicing and defines procedures as reusable services to enhance flexibility and can be easily extended. It minimizes dependencies between the Access Network (AN) and the Core

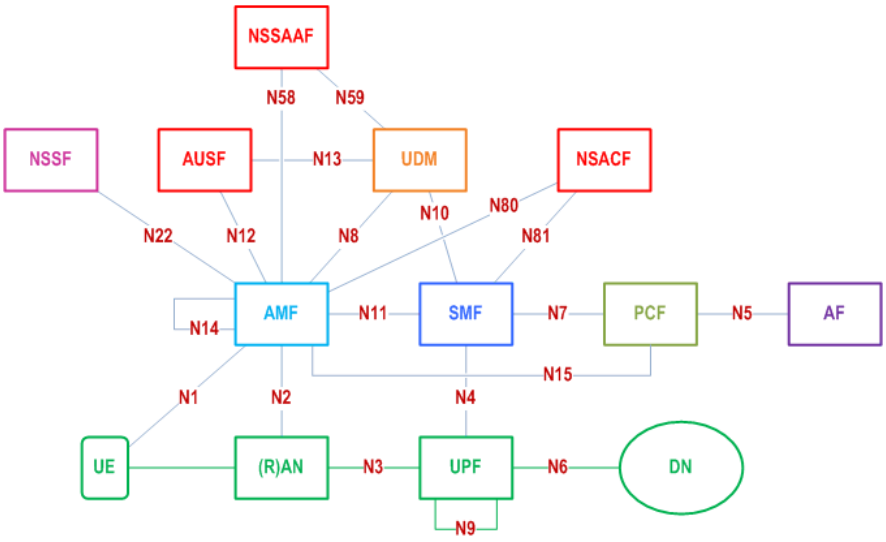


Figure 2.3: Non-Roaming 5G System Architecture

Network (CN) by integrating different access types, including 3GPP and non-3GPP, through a converged CN.

The system includes a unified authentication framework, supports stateless Network Functions (NFs) by decoupling compute and storage resources, and enables capability exposure for network features. It allows concurrent access to local and centralized services and deploys UP functions near the Access Network to support low latency services and local data network access. Additionally, it supports roaming with both home-routed and local breakout traffic in visited networks, ensuring efficient and flexible operation [2].

In 5G, the security framework is built around a new way of organizing the network, known as Service Based Architecture (SBA). This setup introduces new entities [11] and processes that focus on keeping the network secure, especially when it comes to authentication, which is the process of verifying users and devices.

- One of the key entities is the Security Anchor Function (SEAF), which is part of the Access and Mobility Management Function (AMF), located in the serving network, acting as an intermediary during the authentication process [12]. The SEAF receives authentication requests from a device (UE), but it relies on the home network to decide whether the authentication is valid or not. It can reject the authentication, but the final decision rests with the home network.
- The Authentication Server Function (AUSF) [13] is the entity in the home network that actually decides whether the device should be allowed into the network. The AUSF looks at the information provided by the device and checks it against the home network's security policies. It then works with other backend services to compute the necessary data and keys needed to authenticate the device, using secure methods like 5G - Authentication and Key Management (5G-AKA) or Extensible Authentication Protocol - Authentication and Key Management (EAP-AKA').
- The Unified Data Management (UDM) [13] is in charge of managing the data involved

in authentication. One of its key roles is managing the Authentication Credential Repository and Processing Function (ARPF), which selects the right authentication method based on the device's identity and the network's policies. It also helps generate the keys and data that the AUSF uses for authentication.

- Finally, the Subscription Identifier De-concealing Function (SIDF), which is part of the UDM, helps protect the SUPI. In 5G, this permanent identity, which could be a user's IMSI, is always kept hidden and encrypted when sent over the air to prevent hackers from tracking it. The SIDF is the only part of the network that can decrypt the encrypted identity [14] (called the SUCI) using a private key, ensuring that no one else can access the user's personal details.

At its core, this framework introduces a unified and flexible authentication system that seamlessly integrates both 3GPP (traditional cellular) and non-3GPP (such as Wi-Fi or cable) networks. This cross-network compatibility is crucial for enabling a wide range of access methods and supporting the growing ecosystem of connected devices.

Central to this framework is the EAP, which facilitates secure communication between the UE and the AUSF. The SEAF acts as an intermediary, relaying authentication messages between the UE and AUSF [15]. This setup supports various authentication methods, including 5G-AKA, EAP-AKA', and EAP-TLS, providing robust security for data exchange.

2.2.1 Comparing 5G-AKA, EAP-AKA' and EAP-TLS

The authentication process begins authentication method selection. When the SEAF receives a request from the UE seeking network access (see Figure 2.4). The UE provides either a 5G Globally Unique Temporary Identifier (5G-GUTI) or a SUCI to begin the authentication. The AUSF first ensures that the requesting SN is legitimate, then it sends an authentication request to the UDM/ARPF. If the SUCI is provided, the SIDF decrypts it to obtain the SUPI, which is used to determine the authentication method [16].

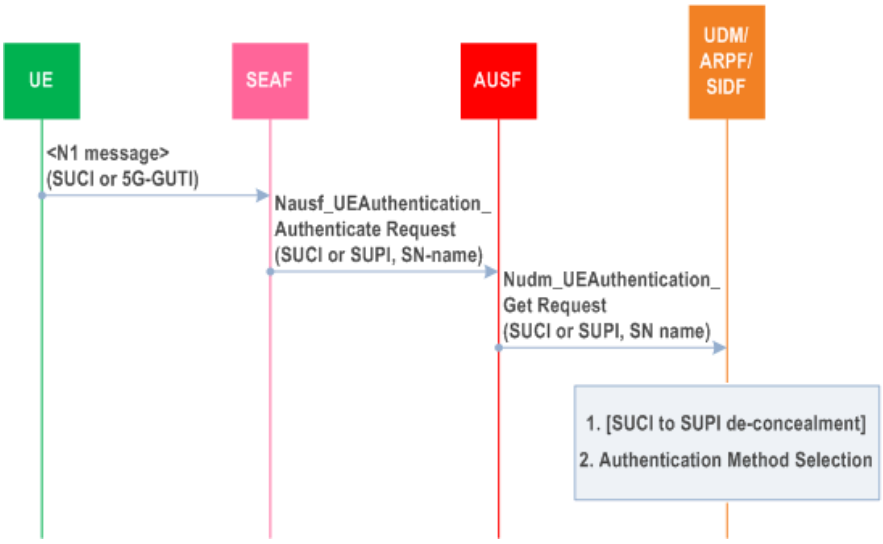


Figure 2.4: Initiation of authentication procedure and selection of authentication method

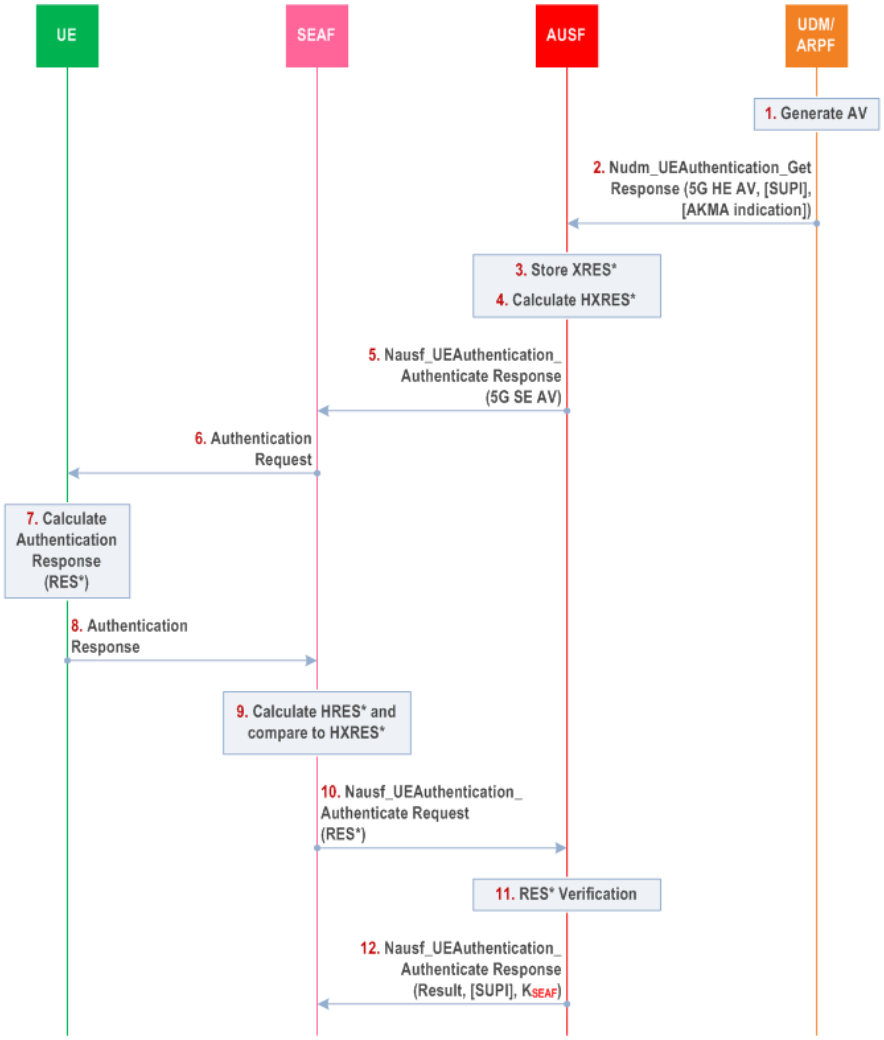


Figure 2.5: Authentication procedure for 5G-AKA

In the case of 5G-AKA, in the next step the UDM/ARPF generates an authentication response containing tokens and keys (see Figure 2.5 message 1). These are sent to the AUSF(message 2), which computes a hash ($HXRES$) and checks the expected response (message 3). The AUSF sends the authentication result, including the $AUTH$ token and $HXRES$, to the SEAF (message 5), ensuring that the SUPI is not exposed to the SEAF, preserving privacy. The SEAF forwards the $AUTH$ token to the UE (message 6), which then validates it using a secret key shared with the home network (message 7). If successful, the UE computes a RES token and sends it back to the SEAF (message 8). The SEAF calculates the $HRES$ and compares it (message 9), and then forwards this to the AUSF (message 10), which validates the response (message 11).

Once the RES token is verified, the AUSF sends an anchor key to the SEAF (message 12). The SEAF derives an AMF key, which the Access and Mobility Management Function uses to generate further keys for securing signaling messages between the UE and network elements. The UE, using its root key, can derive all necessary keys for secure communication with the network, ensuring mutual trust and security [17].

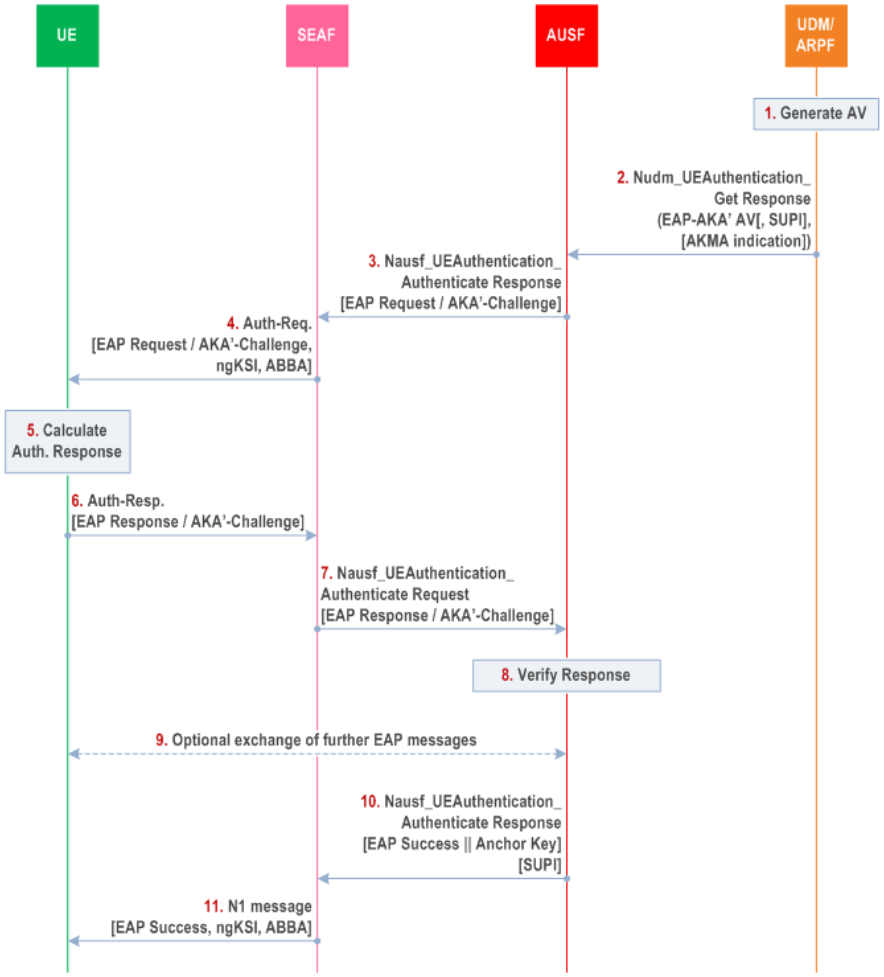


Figure 2.6: Authentication procedure for EAP-AKA'

An alternative authentication method in 5G is EAP-AKA', which provides mutual authentication between the UE and the network using a shared cryptographic key (see Figure 2.6). Unlike 5G-AKA, EAP-AKA' uses EAP messages within NAS messages between the UE and SEAF, and between the SEAF and AUSF. In EAP-AKA', the SEAF merely relays messages between the UE and the AUSF without making authentication decisions. In contrast, in 5G-AKA, the SEAF verifies the UE's authentication response and can act on failures. The K_{AUSF} key in 5G-AKA is generated by the UDM/ARPF and sent to the AUSF, while in EAP-AKA', the AUSF derives this key from the Extended Master Session Key (EMSK), which is provided by UDM/ARPF [18].

Additionally, EAP-TLS (see Figures 2.7 and 2.8) is another optional authentication method suitable for specific scenarios such as private networks or Internet of Things (IoT) devices. Like EAP-AKA', EAP-TLS involves mutual authentication via public key certificates or a Pre-Shared Key (PSK). The SEAF acts as an EAP authenticator, forwarding EAP-TLS messages between the UE and the AUSF. This method differs from the AKA-based approaches by relying on public key certificates for trust, eliminating the need for symmetric keys shared between the UE and the network. This reduces key management risks and does not require a

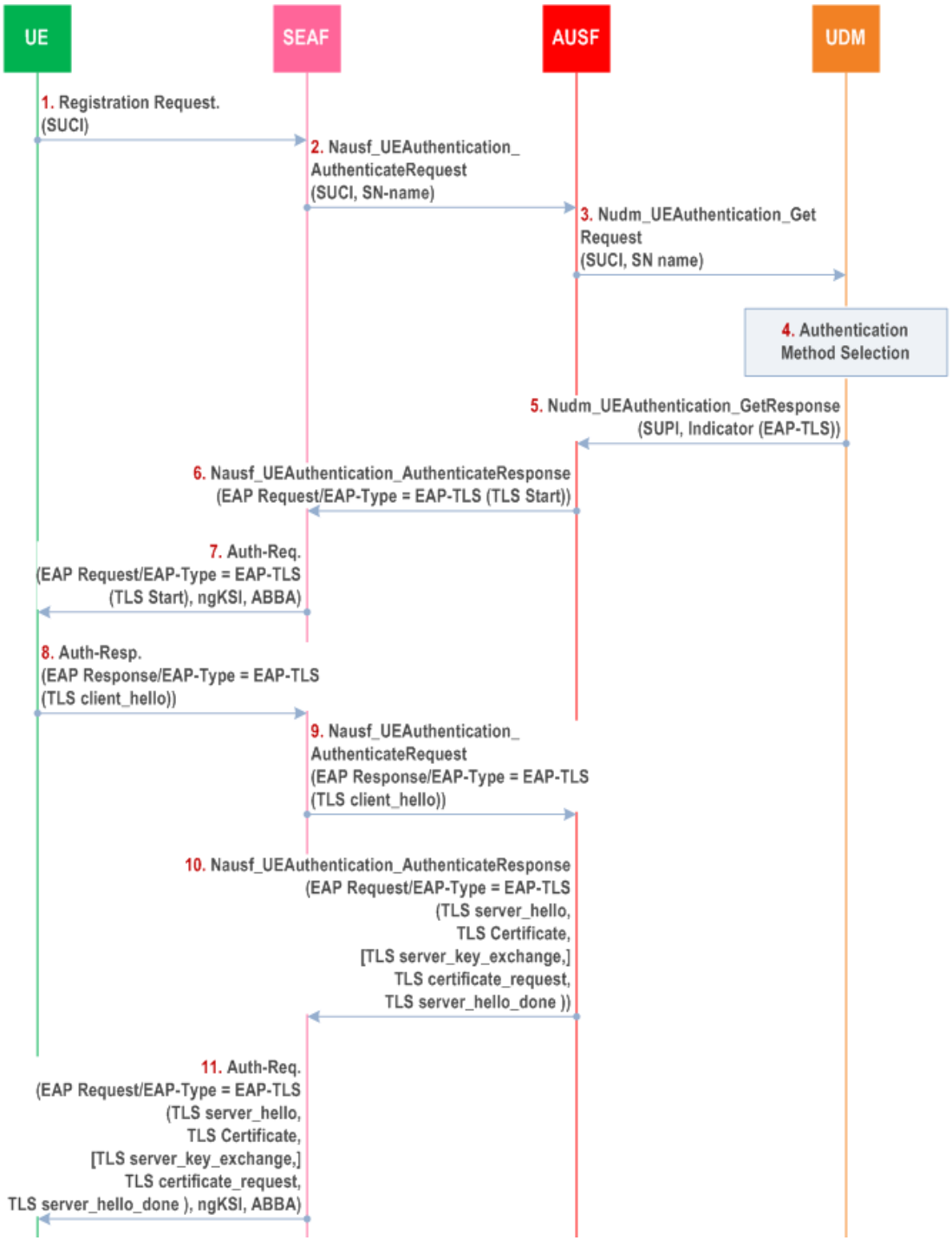


Figure 2.7: Using EAP-TLS Authentication Procedures over 5G Networks for initial authentication (Part 1)

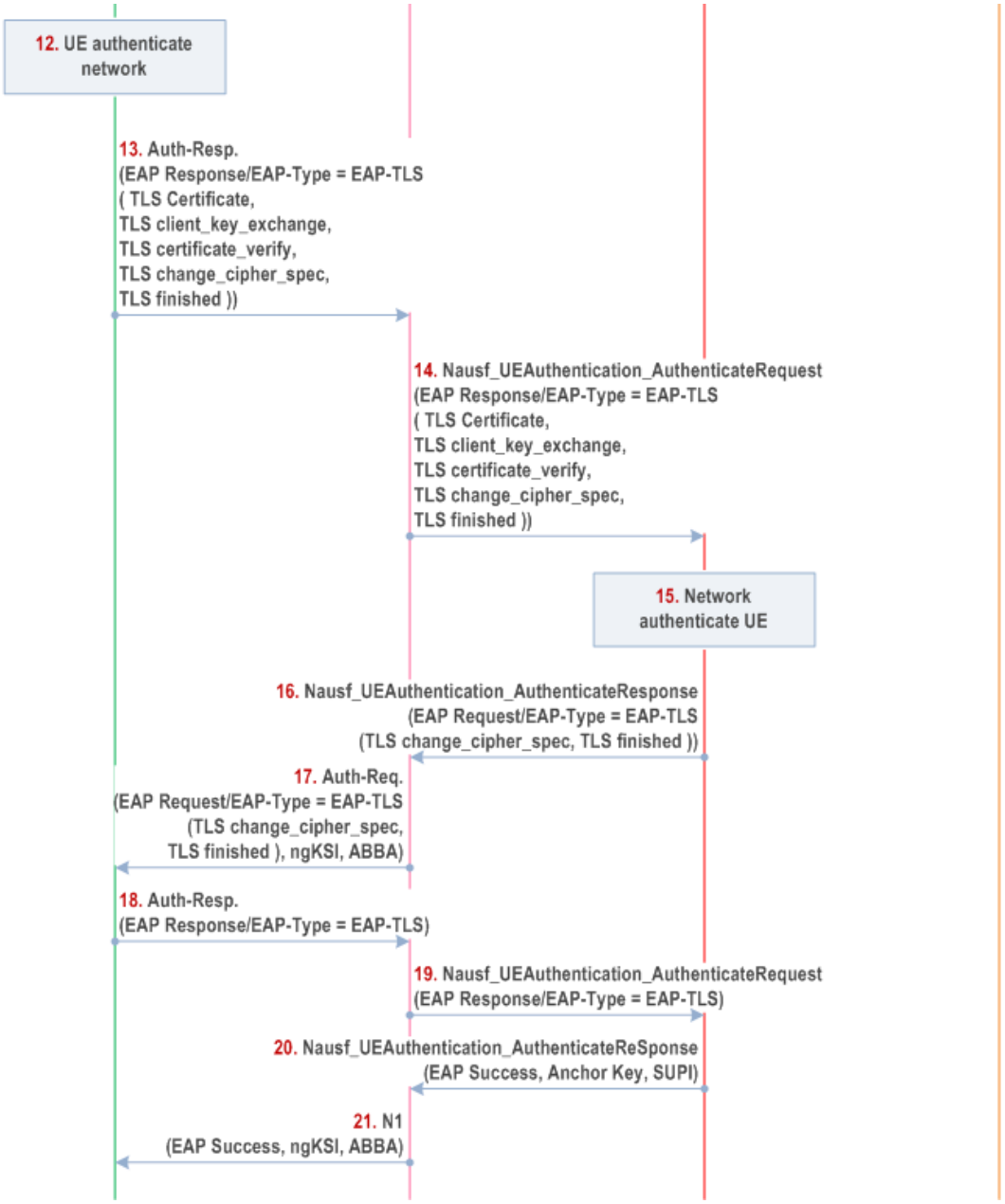


Figure 2.8: Using EAP-TLS Authentication Procedures over 5G Networks for initial authentication (Part 2)

traditional USIM, although secure elements are still needed for storing credentials [19].

2.3 Identity Management in 5G

In the transition to 5G, new mechanisms were introduced to address the vulnerabilities associated with exposed identifiers, such as the IMSI (see Figure 2.9), during RAN communication. These enhancements ensure privacy, security, and compatibility with legacy systems.

One of those mechanisms is the SUPI, which serves as the globally unique identifier for each subscriber within the 5G system. Designed for authentication and provisioning, the SUPI maintains compatibility with legacy formats such as the IMSI and NAI [20]. This flexibility ensures seamless interworking with older systems, including the Evolved Packet Core (EPC).

The SUPI is typically structured as follows:

- IMSI-based SUPI: Includes the Mobile Country Code (MCC), the Mobile Network Code (MNC), and the Mobile Subscriber Identification Number (MSIN) [21].
- NAI-based SUPI: Uses an NAI format (`username@realm`), offering support for scenarios requiring integration with external identity systems or non-3GPP access.

It is important to note that for interworking with EPC, the SUPI must be IMSI-based, ensuring compatibility with existing LTE systems and infrastructure.

Unlike its predecessor, the SUPI is never transmitted in plaintext over the air. Instead, it is concealed as a SUCI (see Figure 2.10) using an Elliptic Curve Integrated Encryption Scheme (ECIES) and the home network's public key. This encryption ensures the confidentiality of user identities during initial registration and subsequent communications.

The SUCI construction includes:

- **SUPI Type**: A value in the range 0 to 7. It identifies the type of the SUPI concealed in the SUCI. These could be an IMSI, a Network Specific Identifier (NSI), a Global Line Identifier (GLI) or Global Cable Identifier (GCI)
- **Home Network Identifier**: Includes the MCC and MNC for routing purposes.
- **Routing Indicator**: 1 to 4 decimal digits, assigned by the home network operator and provisioned in the USIM, that allow together with the Home Network Identifier to route network signalling with SUCI to AUSF and UDM instances capable to serve the subscriber.

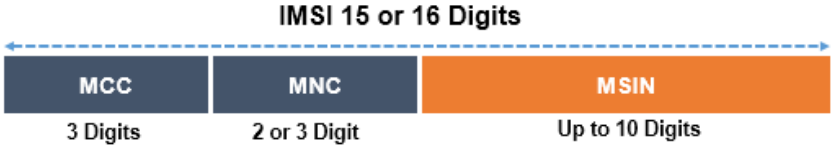


Figure 2.9: IMSI

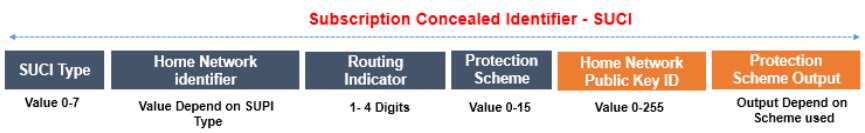


Figure 2.10: SUCI

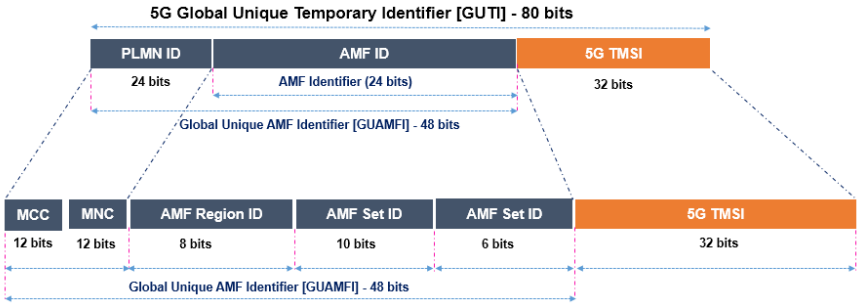


Figure 2.11: 5G-GUTI

- **Protection Scheme ID**: Specifies the encryption method used.
- **Home Network Public Key ID**: Identifies the key applied for encryption.
- **Encrypted Scheme Output**: Represents the concealed SUPI.

The SUCI computation is determined by the operator's policy stored in the USIM. Depending on the configuration, the SUCI may be calculated directly by the USIM or delegated to the Mobile Equipment (ME) [22].

An anonymous SUCI is composed by setting the SUPI Type field to 1 (Network-Specific Identifier), using the null protection scheme, and where the scheme output corresponds to a username set to either the "anonymous" string or to an empty string.

To further enhance privacy, 5G utilizes temporary identifiers during communication. The 5G Globally Unique Temporary Identifier [23](see Figure 2.11) is dynamically assigned by the AMF and replaces the SUPI in subsequent signaling exchanges. This frequent reassignment minimizes the risk of user tracking.

The 5G-GUTI is typically in a format comprising:

1. **Globally Unique AMF Identifier (GUAMI)**: Identifies the AMF managing the UE's session.
2. **5G Temporary Mobile Subscriber Identity (5G-TMSI)**: Uniquely identifies the UE within the AMF context.

For efficient radio signaling, a shortened version, the 5G Short Temporary Mobile Subscriber Identity (5G-S-TMSI), is utilized. The 5G-S-TMSI is the shortened form of the GUTI to enable more efficient radio signalling procedures (e.g. during Paging and Service Request)

Additionally, the 5G-GUTI can be represented in an NAI format when required [24]. This flexibility supports interworking and ensures compatibility across diverse network scenarios.

The AMF retains the flexibility to assign new 5G-GUTI values at any time, though updates are generally synchronized with the next NAS signaling exchange to avoid unnecessary interruptions. Despite these mechanisms, scenarios such as initial network access or failure to resolve a temporary identifier necessitate direct use of the SUPI.

In addition to subscriber identifiers, the Permanent Equipment Identifier (PEI) uniquely distinguishes user equipment capable of accessing the network. The PEI is critical for device management but is safeguarded to prevent unauthorized tracking [20].

The PEI adheres to specific formats based on device type and use case:

- For devices supporting 3GPP access, the International Mobile Equipment Identity (IMEI) format is mandated, ensuring uniformity.
- The PEI is presented with an indication of its format, enabling compatibility across diverse use cases.

2.4 Access Network Types in 5G

2.4.1 3GPP vs non-3GPP

3GPP encompasses standards for mobile networks like third-generation (3G), 4G, and 5G, which are cellular technologies enabling network services from mobile carriers. These networks operate on licensed spectrum, ensuring predictable performance, security, and quality of service.

In contrast, non-3GPP access refers to technologies not standardized by 3GPP, such as Wi-Fi or satellite networks. These networks operate on unlicensed or partially licensed spectrum, are typically managed by different standards bodies (e.g., IEEE for Wi-Fi), and are widely used for cost-effective and ubiquitous connectivity [25]. While non-3GPP networks were previously considered external to mobile networks, 5G allows their tighter integration into the core network, enabling seamless user experiences across both network types.

5G introduces the capability to support communication across both 3GPP and non-3GPP access networks. This integration extends beyond traditional cellular devices, allowing a wide range of UEs and non-UEs devices—such as IoT sensors, laptops, and legacy equipment—to connect securely and efficiently [26].

5G supports communication across 3GPP and non-3GPP access networks using distinct architectures for trusted and untrusted access. Trusted non-3GPP networks rely on the Trusted Non-3GPP Gateway Function (TNGF) as seen in Figure 2.12, while untrusted networks leverage the Non-3GPP Interworking Function (N3IWF) as seen in Figure 2.13. Both gateway functions connect to the 5GC's control and user planes via the N2 and N3 reference points.

When using non-3GPP access, UEs establish secure IP security (IPSec) tunnels with the N3IWF or TNGF to register with the 5GC. Post-registration, the NAS signaling between

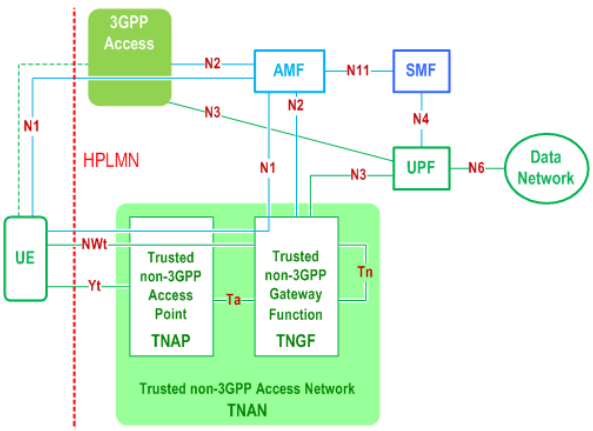


Figure 2.12: Architecture for 5GC with Trusted Non-3GPP Access

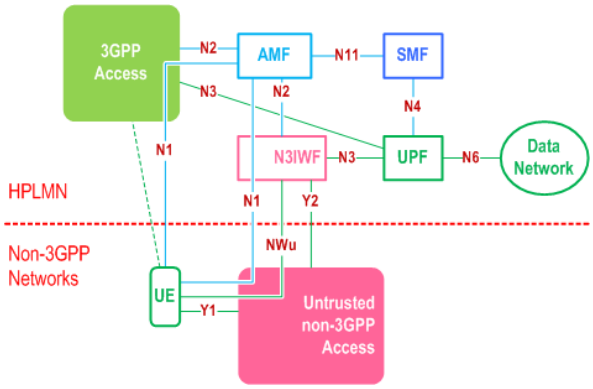


Figure 2.13: Architecture for 5GC with Untrusted Non-3GPP Access

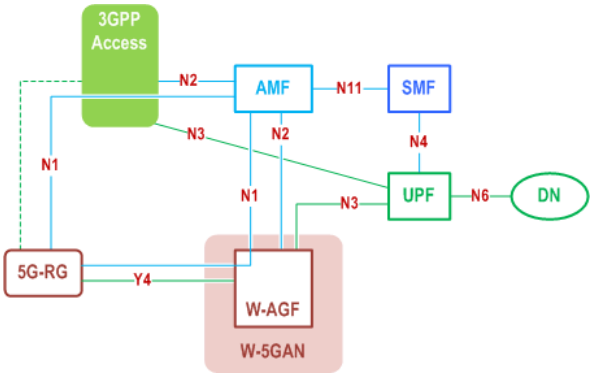


Figure 2.14: Architecture for 5GC for 5G-RG with W-5GAN and NG-RAN

the UE and the core network is protected using the same security mechanisms, N1, as 3GPP access.

5G also enables operators to merge fixed and mobile networks through functional convergence, offering a unified control plane for both wireline and wireless sessions. This approach enhances seamless access-independent services, supports multi-access connectivity, simplifies network operations, unifies technology and training, centralizes subscriber management, extends 5G core coverage, and broadens fixed access service offerings [27].

Wireline 5G Access Network (W-5GAN), such as broadband fiber-optic networks, connects to the 5GC via the Wireline Access Gateway Function (W-AGF) (see Figure 2.14), using N2 and N3 interfaces for control and user plane functions, respectively. When a 5G Residential Gateway (5G-RG), such as a home router with 5G capabilities, connects through both Next Generation Radio Access Networks (NG-RAN) , like a 5G cellular tower (e.g. a Fixed Wireless Access (FWA) context), and W-5GAN, it maintains separate N1 signaling instances for each access. However, a single AMF in the same 5GC serves the 5G-RG. NAS signaling over W-5GAN persists even after Protocol Data Unit (PDU) sessions are released [28].

2.4.2 Device Diversity and Access Options

As the 5G network evolves, it's important to recognize that not all devices connected to the network are 5G capable. While we typically envision UE as being 5G-enabled, 3GPP has also accounted for a wide range of devices, from legacy systems to non-5G capable ones,

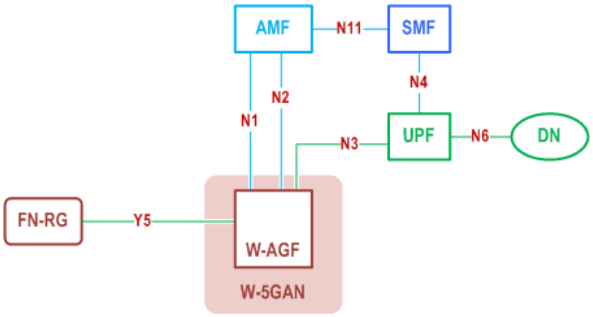


Figure 2.15: Architecture for 5GC for FN-RG with W-5GAN and NG-RAN

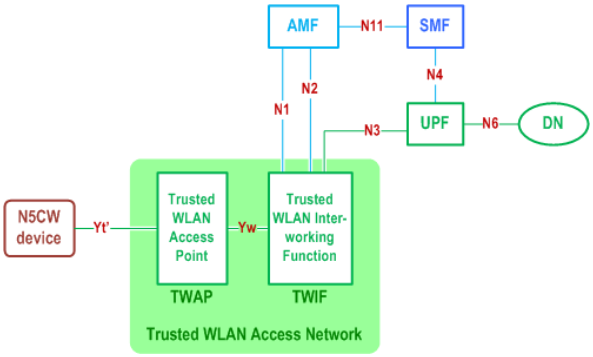


Figure 2.16: Architecture for supporting 5GC access from N5CW devices

ensuring that connectivity remains seamless and secure across diverse access points.

For Fixed Network Residential Gateways (FN-RGs), such as legacy home routers, connected via W-5GAN (see Figure 2.15), the W-AGF handles N1 signaling on behalf of the FN-RG. UEs, like smartphones or IoT devices, connecting through these gateways can access the 5GC via either N3IWF (untrusted access using Wi-Fi) or TNGF (trusted access) depending on the network configuration.

There are also devices that are not 5G-capable over WLAN access [29], referred to as Non-5G Capable over WLAN (N5CW) devices, cannot support 5GC NAS signaling over WLAN but may still operate as 5G UEs over NG-RAN. 3GPP provides enhancements for N5CW devices to access 5GC via trusted WLAN access networks (see Figure 2.16), which are a type of Trusted Non-3GPP Access Network (TNAN), typically using IEEE 802.11 technology. These networks must support specific functions, such as the Trusted WLAN Interworking Function (TWIF), which enables N5CW devices to register with the 5GC. When a N5CW device performs a EAP-based access authentication procedure to connect to a trusted WLAN access network, it may simultaneously be registered to an 5GC of a Public Land Mobile Network (PLMN) or Service Network Public Land Mobile Network (SNPN). The TWIF handles authentication, AMF selection, NAS protocol communication, and relays user data between the WLAN access network and the 5GC. In this specification, trusted WLAN access for N5CW devices only supports IP PDU sessions.

2.4.3 Authentication Flow Across Trusted and Untrusted Networks

Examining the authentication flows for devices connecting to the 5GC via non-3GPP access networks reveals differences in the mechanisms used for trusted and untrusted accesses.

For untrusted non-3GPP access, security is established using Internet Key Exchange version 2 (IKEv2) to set up IPSec security associations between the UE (acting as the Internet Key Exchange (IKE) initiator) and the N3IWF (acting as the IKE responder). The UE and N3IWF use a derived key from the AMF to complete the authentication process.

In non-roaming scenarios, the home operator (or Home Operator Public Land Mobile Network (HPLMN)) decides whether a non-3GPP access network is trusted or untrusted based on its security features, while in roaming scenarios, the decision is made by the UDM in the HPLMN. This decision applies consistently across all Data Networkss (DNs) the UE connects to via the same non-3GPP access network.

The UE stores trusted non-3GPP access network information in the USIM, which takes priority over the ME, the device itself.

For authentication over untrusted non-3GPP networks (see Annex B), the UE uses a vendor-specific EAP method called "EAP-5G", which employs the "Expanded" EAP type and the 3GPP Vendor-Id. The Extensible Authentication Protocol 5G (EAP-5G) method is used between the UE and N3IWF to encapsulate NAS messages. If the UE requires authentication by the 3GPP home network, standard authentication methods are applied between the UE and the AUSF. Whenever possible, the UE will reuse the existing NAS security context from the AMF for authentication [30].

Security for trusted non-3GPP access to the 5GC involves the UE registering to the 5GC via a TNAN using the EAP-5G procedure, similar to that used for untrusted access (see Appendix C). The link between the UE and the TNAN relies on Layer-2 security, making IPSec encryption unnecessary between the UE and the TNGF, though integrity protection is ensured [31].

During registration, the TNGF terminates EAP-5G signaling and forwards NAS messages to the 5GC. At the registration's conclusion, an IPSec Security Association Network Termination (IPSec SA (NWt)) (an IPSec Security Association managing secure communication parameters at a Network Termination point) is established between the UE and TNGF to protect NAS messages. Additional IPSec Security Associations (IPSec SAs) are created during PDU session establishment for user plane transport. Security policies, determined by the home operator, define whether non-3GPP access is trusted based on security domains or other considerations.

For trusted non-3GPP access authentication, key differences from untrusted access include avoiding IKEv2 encapsulation for EAP-5G packets, utilizing 5G-GUTI or SUCI for UE identity, and deriving keys like K_TNGF and K_TNAP for secure communication. These keys are shared between the AMF, TNGF, and Trusted Non-Access Stratum Proxy (TNAP) to establish secure communication flows.

2.5 Device Support Behind Wireline

As we shift focus from non-3GPP access networks, it's important to examine wireline access, which is crucial for connecting home devices like laptops, smart TVs, and IoT devices to the 5GC. Many of these devices rely on wireline connections, often through home routers like 5G-RG and FN-RG [32], to access the 5G network. Understanding how wireline access works is vital, as it bridges the gap for devices without full 5G capabilities, ensuring seamless integration across both fixed and wireless environments.

For instance, in a smart home, a 5G-RG might connect to the 5GC using fiber or FWA, while an enterprise could use an FN-RG for secure, high-speed access via wireline networks. The connection between the Residential Gateway (RG) and W-5GAN leverages 5G security frameworks, similar to wireless setups. However, roaming is not supported for these entities or the devices they serve. In specific configurations, additional EAP methods can be used to ensure secure authentication, enabling devices like remote workstations to securely access the 5GC. This enables seamless and secure integration of 5G across fixed and wireless environments.

The 5G-RG supports 5GC connections via NG-RAN, W-5GAN, or both, with registration processes varying by access type [33] [34]. As the 5G-RG is treated as a UE by the 5GC, it uses the standard authentication framework, including 5G-AKA and EAP-AKA'. For W-5GAN connections, Wireline Control Plane (W-CP) protocol stack messages encapsulate NAS signaling. In contrast, the FN-RG connects exclusively via W-5GAN and relies on the W-AGF to handle N1 signaling on its behalf. The W-AGF provides connectivity to the 5GC via N2 and N3 interfaces and can authenticate the FN-RG based on local policies. Secure protocols like Network Domain Security/IP (NDS/IP) or Datagram Transport Layer Security (DTLS) establish mutual trust between the wireline operator managing the W-5GAN and the PLMN operator managing the 5GC [35].

Devices such as N5CW occupy a middle ground. While they lack full 5G functionality over WLAN, they can still register with the 5GC, establish PDU sessions, and authenticate using 3GPP credentials (USIM). These devices may function as regular UEs when connected via cellular networks, unlike N5GC devices, which lack 5G-specific capabilities [36].

In wireline access, gateways like the 5G-RG and FN-RG enable connectivity for devices that cannot independently handle NAS signaling or derive 5G keys. For example, the W-AGF manages registration on behalf of FN-RG devices and ensures secure communication with the 5GC. For N5GC devices connecting via Customer Residential Gateway (CRG), the SUPI includes a network-specific identifier (NAI), and the W-AGF derives the SUCI from the EAP-Identity message, passing it to the AMF for secure identification [37].

This architecture enables a diverse range of devices, from IoT sensors to legacy systems, to benefit from 5G connectivity without requiring full 5G capabilities. It highlights the critical role of wireline access in achieving seamless convergence across fixed and wireless network environments while maintaining robust security.

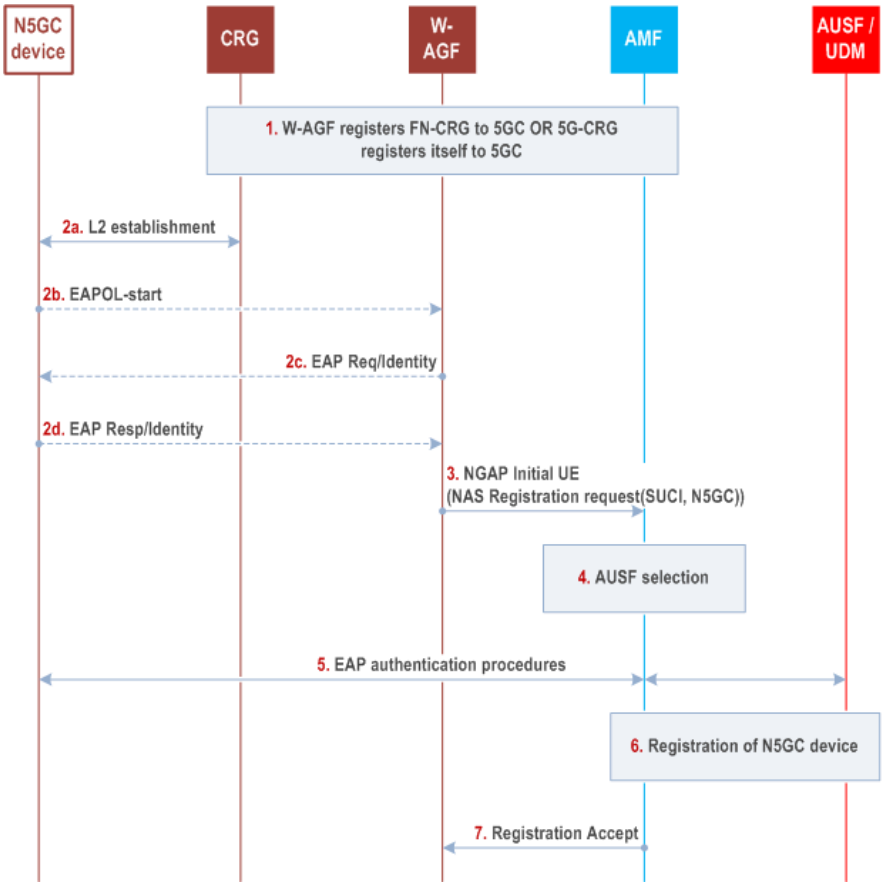


Figure 2.17: 5GC registration of N5GC device

2.5.1 5GC Registration Process for N5GC Devices

In isolated 5G networks with wireline access, N5GC devices can access the 5GC through a structured process involving EAP-based authentication. Each N5GC device is treated as an individual entity with its own subscription record in the UDM/Unified Data Repository (UDR), distinct from the subscription record of the CRG. The CRG operates in L2 bridge mode, forwarding traffic from connected N5GC devices to the W-AGF for further processing and registration.

The process begins with the registration of the CRG to the 5GC (the flow is shown in Figure 2.17). This enables the CRG to act as a bridge, facilitating communication between N5GC devices and the W-AGF. Once this setup is in place, authentication is triggered when the CRG forwards traffic from an N5GC device. This occurs either through the reception of an *Extensible Authentication Protocol Over LAN (EAPOL)-Start* frame sent by the N5GC device or when the W-AGF detects traffic from an unknown MAC address. The N5GC device responds by sending an *EAP-Response/Identity* message containing its NAI, formatted as `username@realm`.

The W-AGF then acts on behalf of the N5GC device to initiate its registration with the 5GC. It constructs and sends a *NAS Registration Request* to the AMF, including a SUCI derived from the NAI. This registration explicitly indicates that the device lacks

native 5G capabilities. The W-AGF establishes separate Next Generation Access and Service Management Protocol (NGAP) connections for each N5GC device over the N2 interface, enabling distinct communication channels for every device.

Authentication of the N5GC device is carried out by the AUSF using EAP-based methods. Once the device successfully authenticates, the AUSF provides the relevant security information to the AMF, including the SUPI derived from the NAI. This SUPI uniquely identifies the N5GC device within the 5GC ecosystem, ensuring individual accountability and secure operation.

Following successful authentication, the AMF completes additional registration procedures. If a PEI is required, the W-AGF uses the MAC address of the N5GC device, with an option to encode it in IEEE Extended Unique Identifier-64 (EUI-64) format depending on operator policy. Once registration is finalized, the W-AGF communicates the *Registration Accept* message to the N5GC device, marking the completion of the process.

After registration, the W-AGF establishes a single PDU session for each N5GC device, ensuring each device is assigned its own unique data session within the 5GC while accounting for the device's limitations. This ensures secure and individualized connectivity. Additionally, the W-AGF manages NGAP connections, ensuring that if the NGAP connection for a CRG is released, all associated N5GC device connections are also terminated. The CRG continues to operate as an Fixed Network Customer Residential Gateway (FN-CRG), supporting seamless communication for connected devices [38].

In Annex 0 of TS 33.501, we can get more detail regarding this registration and authentication process (see Appendix D) [36].

2.5.2 N5GC and NAUN3 devices

A Non-Authenticable Non-3GPP (NAUN3) device does not support NAS signalling, is connected to 5GC via a RG and does not support authentication with the 5GC [39].

NAUN3 devices, which cannot be authenticated by the 5GC, may be locally authenticated by the 5G-RG using methods like pre-shared secrets. Examples of pre-shared secrets include Wi-Fi passphrases for Service Set Identifiers (SSIDs), Personal Identification Number (PIN) codes, or static security keys configured during device setup. Differentiated services, including Quality of Service (QoS) and network slicing, can be applied to these devices through Connectivity Groups Identifiers (CGIDs) (see Figure 2.18) [40].

Each CGID corresponds to a specific physical or virtual port on the 5G-RG, such as Ethernet ports, WLAN SSIDs, or Virtual Local Area Networks (VLANs). Devices connected to the same logical port are considered part of the same CGID, and each CGID maps to a separate PDU Session established by the 5G-RG to manage their traffic.

The 5G-RG is configured with port information, such as VLANs and SSID, via standardized protocols like TR-69, TR-360, and TR-181. User Registration Service Protocol (URSP) rules are provided to the 5G-RG to define how CGIDs are mapped to PDU Session parameters, such as the Data Network Name (DNN) and Single Network Slice Selection Assistance Information (S-NSSAI). These mappings determine how traffic is routed and which network

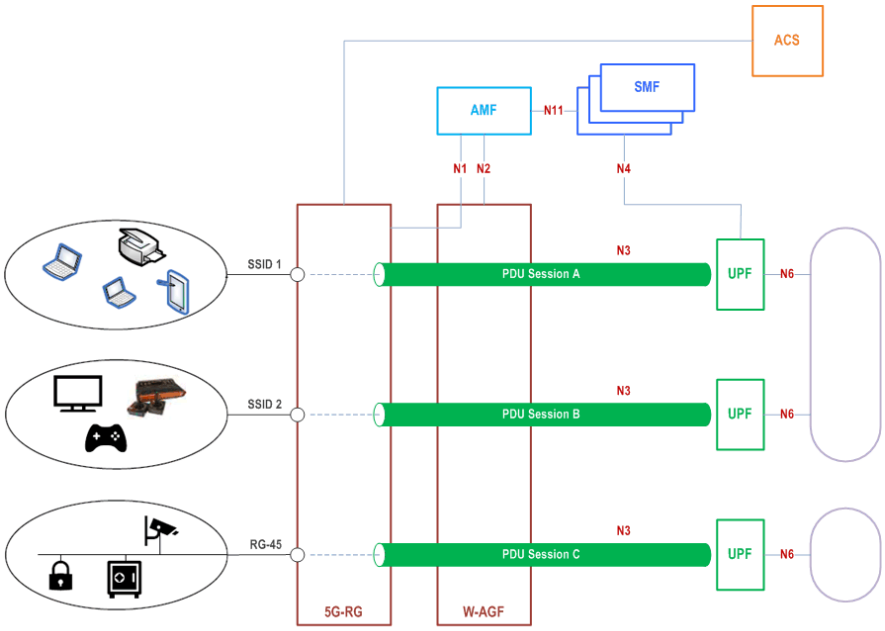


Figure 2.18: NAUN3 devices behind 5G-RG based on connectivity groups

slice the devices use. For instance, a home office CGID might map to a DNN providing enterprise services and an S-NSSAI prioritizing low latency for work-related tasks.

Charging and QoS differentiation for NAUN3 devices can be implemented through Policy and Charging Control (PCC) rules. These rules define service flows tied to specific PDU Sessions, enabling detailed traffic management and billing policies. Additionally, isolation of devices using a specific CGID into a separate network slice (associated with an S-NSSAI) can provide enhanced security and service customization. For example, devices in a child's CGID could be isolated into a network slice with strict content filtering and bandwidth limitations [40].

The main difference between NAUN3 and N5GC devices lies in their capabilities and how they interact with the 5GC, here is a summary of what we've seen soo far (also represented in Table A.1 in Appendix A :

- **NAUN3 Devices**
  - **Authentication**: They cannot be authenticated by the 5GC. Instead, they rely on local authentication mechanisms provided by the 5G-RG (e.g., Wi-Fi passphrases, PINs, or pre-shared keys).
  - **Connection**: NAUN3 devices connect through the 5G-RG, which maps their traffic to PDU Sessions and handles aspects like QoS and network slicing (e.g., via S-NSSAI).
  - **Subscription Records**: NAUN3 devices do not have subscription records in the 5GC and operate entirely under the configuration and policies of the 5G-RG.
  - **Purpose**: They are typically legacy or IoT devices that do not need direct 5GC access but require differentiated services provided via local configuration and mapping, and need to be connected via Layer 2 to 5G devices via extended slices.

  - **Example**: A smart home appliance connected to the 5G-RG via Wi-Fi using a pre-shared key, with its traffic routed through a dedicated network slice.
- **N5GC Devices**
  - **Authentication**: They can be authenticated by the 5GC using EAP-based authentication with the help of the W-AGF, which acts as an intermediary.
  - **Connection**: N5GC devices connect to the 5GC through wireline access (e.g., fiber or Digital Subscriber Line (DSL)) and use the W-AGF to handle their registration, authentication, and session management.
  - **Subscription Records**: Each N5GC device has its own unique subscription record in the UDM/UDR, separate from the subscription record of the CRG.
  - **NGAP Connections**: The W-AGF establishes separate NGAP connections for each N5GC device over the N2 interface to the AMF. This enables individual session and mobility management for each device.
  - **Purpose**: N5GC devices extend 5GC services to fixed network devices that do not possess 5G capabilities.
  - **Example**: A desktop computer connected to the 5GC via fiber access and authenticated using EAP over the W-AGF.

In summary, NAUN3 devices operate entirely locally, with no interaction with the 5GC, while N5GC devices leverage intermediaries like the W-AGF to authenticate and establish PDU Sessions with the 5GC, maintaining unique subscription records and dedicated NGAP connections.

2.5.3 3GPP Advancements on QoS Traffic Differentiation

Recent updates within 3GPP Release 19, particularly in TS 23.316 (Wireless and wireline convergence access support for the 5G System), TS 23.501 (System Architecture), TS 23.502 (Procedures), and TS 23.503 (Policy and Charging Control), have introduced and refined mechanisms for providing differentiated QoS to individual non-3GPP devices operating behind a 5G-RG as ilustrated in Figure 2.19. These enhancements detailed in TS 23.316 clauses 4.10e [33], Annex C [41], and TS 23.501 clause 5.52 [42], signify a focused effort towards granular traffic management and service personalization for devices in converged environments.

Key Developments in Differentiated QoS for Non-3GPP Devices

1. Identification of Individual Non-3GPP Devices
   - A central concept is the "Non-3GPP Device Identifier" a generic string unique within the scope of the 5G-RG's SUPI [43]. This identifier enables the 5G System to distinguish traffic from different non-3GPP devices connected through the same 5G-RG, even if they initially share a PDU Session.
   - The 5G-RG (acting as the UE in this context) may bind this identifier to a specific non-3GPP device, although the exact method of binding and how the 5G-RG becomes aware of these identifiers is considered implementation-specific [43].
2. Provisioning of Device-Specific QoS Information

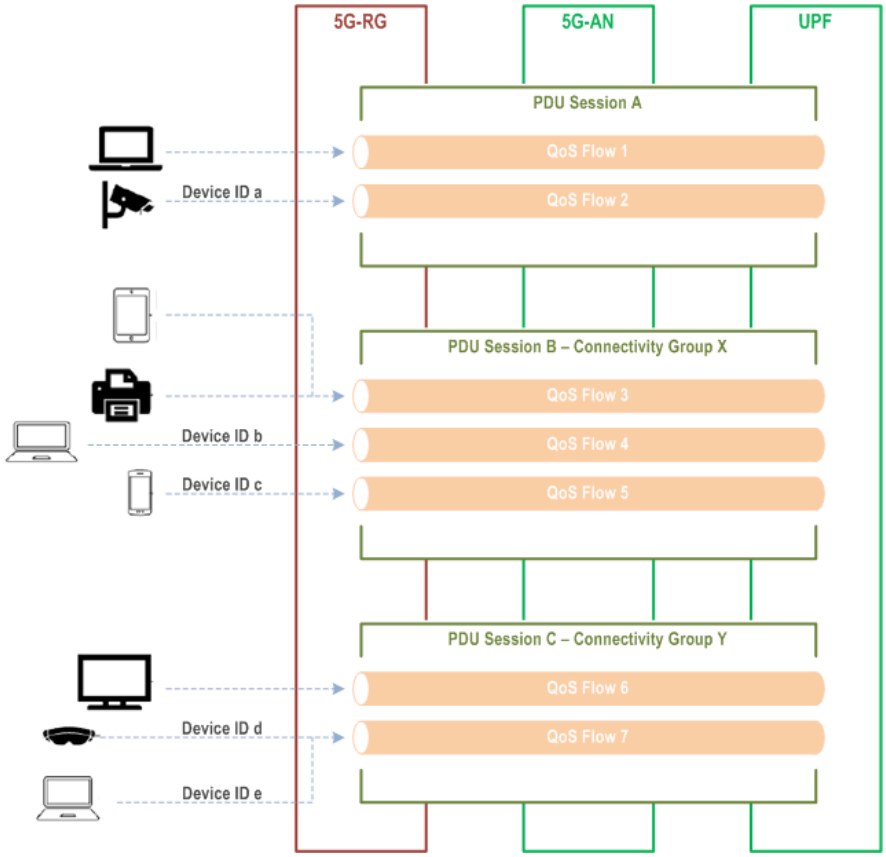


Figure 2.19: Example scenario for mapping traffic of individual non-3GPP devices behind 5G-RG to a PDU Session

- An Application Function (AF), which can belong to the operator or a third party, can provision "Non-3GPP Device Identifier Information" into the UDR via the Network Exposure Function (NEF) [42].
- This information includes the Non-3GPP Device Identifier, associated QoS parameters (either a QoS reference or individual parameters), and optionally DNN/S-NSSAI or flow descriptions. The NEF uses an AF-Service-Identifier to recognize the request type and sets the Data Subset Identifier to "Non-3GPP Device Identifier Information" when interacting with the UDR. This provisioning is typically done for devices requiring specific QoS treatment.

3. Session Management and Policy Control Enhancements
   - The 5G-RG can signal "Non-3GPP Device Connection Information" (which includes the Non-3GPP Device Identifier, MAC address, and optionally VLAN ID or Internet Protocol (IP) address/port ranges for different PDU session types) to the Session Management Function (SMF) via a PDU Session Modification Request [43].
   - The SMF forwards this information to the Policy Control Function (PCF) [44]
   - The PCF, upon receiving the Non-3GPP Device Identifier and its user plane address, creates or modifies PCC rules. It determines the appropriate QoS parameters based on the Non-3GPP Device Identifier Information retrieved from the UDR and/or operator policy. These PCC rules are then installed at the SMF.

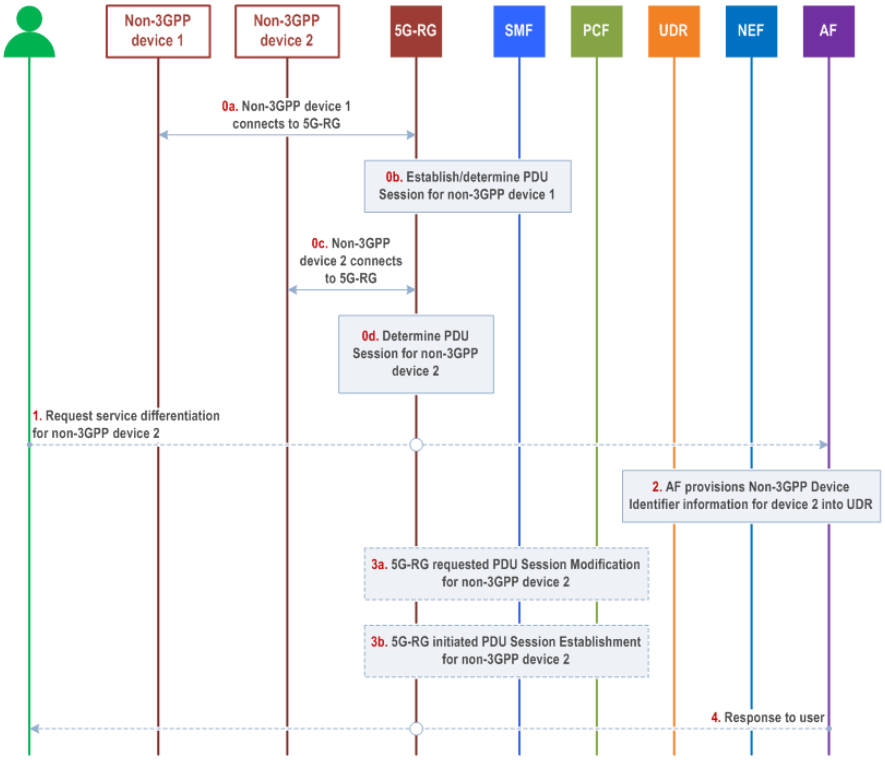


Figure 2.20: Registration flow for an example scenario for mapping traffic of individual non-3GPP devices behind 5G-RG to a PDU Session

- If the PCF finds that a provided Non-3GPP Device Identifier is not available for the 5G-RG's subscription, it informs the SMF, which then rejects the PDU Session Modification. Similarly, if an identifier is removed from the subscription, the UDR notifies the PCF, which then initiates the removal of associated PCC rules.

4. Procedural Framework for Differentiated Service Activation [33] [41]

- As shown in Figure 2.20, initially a non-3GPP device connecting to the 5G-RG might have its traffic mapped to a PDU Session based on URSP rules (potentially using a Connectivity Group ID [40]) or local configuration. Multiple devices might share this PDU Session and its default QoS Flow.
- A request for differentiated QoS for a specific device (e.g., device 2) can be initiated by the 5G-RG subscription owner via an AF (e.g., an operator portal).
- The AF then provisions the Non-3GPP Device Identifier Information for device 2 into the UDR (step 2 in Figure 2.20).
- This triggers (Step 3 of Figure 2.20) either a 5G-RG requested PDU Session Modification or a 5G-RG requested PDU Session Establishment. The choice between modification of an existing PDU Session (to add/modify QoS Flows) or establishment of a new PDU Session for the device appears to be based on network configuration.
- Following this, the traffic of the identified device is mapped to a specific QoS Flow (potentially a new one within an existing PDU session or within a newly established PDU session) that provides the requested differentiated service.

These 3GPP advancements in Release 19 provide a standardized framework for identifying individual non-3GPP devices behind a 5G-RG and applying differentiated QoS policies to their traffic, enhancing service flexibility and network resource management.

CHAPTER 3

Methodology And Proposed Framework

This chapter explains the research approach undertaken to tackle challenges of integrating NAUN3 devices into 5G networks. Outlines the specific methodologies employed for analysis of compatible authentication mechanisms as well as for managing device identity. Furthermore, it presents the designed framework for seamless, secure integration, detailing the motivation behind the architecture as well as essential building blocks.

3.1 Overall Research Approach

This research followed a constructivist methodology to develop a novel solution for integrating NAUN3 devices into 5G networks by repurposing existing 5G facilities and capabilities.

The initial phase involved a detailed study of the 5G architecture and relevant 3GPP standards. Although these standards include mechanisms for non-3GPP access and for managing devices behind residential 5G-RGs, they fall short of addressing the specific challenges posed by NAUN3 devices—particularly in terms of USIM requirements and the inability to perform standard 5G authentication. The core issue identified was enabling 5GC acceptance, indirect authentication, and per-device handling without requiring changes to either the NAUN3 devices or the 5GC itself. Concepts such as CGIDs [40] and PDU session separation were explored for inspiration, although detailed implementation strategies for such contexts are not specified in current standards.

To minimize required changes to both the 5GC and NAUN3 devices, the proposed solution centers on implementing all adaptation logic within the 5G-RG. This approach ensures transparency to both the core network and the end devices, while granting control to the network operator.

The central methodology involved designing a framework in which the 5G-RG acts as a mediator. This framework employs local authentication (specifically EAP-TLS), with the

5G-RG serving as an authenticator that forwards requests to a mandatory, operator-controlled external EAP server. This server authenticates devices lacking native 5G identities. Upon successful local authentication, the 5G-RG establishes a dedicated PDU Session in the 5GC for each authenticated device. This session acts as a 'proxy identity', allowing the 5GC to manage the device's traffic without requiring direct knowledge of its non-5G credentials. The 5G-RG also handles traffic forwarding between the device and its associated PDU Session.

This framework, designed to meet integration requirements with minimal disruption, was subsequently detailed, prototyped in a testbed, and validated through functional and security testing, as elaborated in the following chapters. The approach was selected for its potential to deliver a practical, minimally invasive solution to the integration challenge.

3.2 Requirements Analysis

To address the challenge of integrating NAUN3 devices lacking native 5G credentials, while maintaining minimal disruption to existing systems, the following key requirements were defined to guide the framework design:

1. **Minimal Impact on Existing Infrastructure**
   a) **Core Network:** Standard 5GC functions (AMF, SMF, User Plane Function (UPF)) must remain unaltered at the code level, with only configuration changes (e.g., IP bindings, DNN definitions) permitted.
   b) **End Devices:** NAUN3 devices must operate without hardware or software modifications, requiring only standard EAP Supplicant functionality.
   c) **RAN:** Standard 5G RAN components (Next Generation Node Bs (gNBs)) must function without modification, maintaining standard interface operations with the core.
2. **5G-RG-Centric Intelligence**
   - All adaptation logic must reside within the network's 5G-RG, which mediates between the NAUN3 device and the 5GC, in support of the minimal-impact goal.
3. **Functional Requirements**
   - **Secure Device Onboarding:** Devices must be locally authenticated via EAP-TLS, with the 5G-RG relaying requests to an operator-controlled external EAP server.
   - **Individual Device Representation:** Each authenticated device must be assigned a unique PDU Session within the 5GC, serving as a proxy identity.
   - **Traffic Separation:** Internal service traffic (e.g., Remote Authentication Dial-In User Service (RADIUS) communication) and end-device traffic must be clearly separated within the 5G transport network.
   - **Lifecycle Management:** The 5G-RG must manage each device's session from initial authentication to disconnection, including re-authentication and PDU Session teardown.

- **Traffic Mapping and Isolation:** The 5G-RG must ensure precise mapping and isolation of traffic between each device and its dedicated PDU Session.

4. **Operational Requirements**
   a) **Transparency:** The system must appear standard to both the 5GC (via PDU procedures) and the NAUN3 device (via EAP authentication).
   b) **Operator Manageability:** The complete solution, including 5G-RG logic and external EAP infrastructure, must be operator-deployable and manageable.

Together, these requirements establish the foundation for a solution that integrates unmodified NAUN3 devices into the 5G ecosystem with minimal disruption to existing network components and processes.

3.3 Proposed Authentication Mechanism

To securely onboard NAUN3 devices lacking native 5G credentials, this framework employs EAP-TLS—a mutual certificate-based authentication method offering strong security and compatibility with standard operator-managed components like RADIUS servers. EAP-TLS is particularly suitable for IoT devices, where password-based methods are often impractical or insecure. Figure 3.1 illustrates the architecture, which comprises three roles:

1. **Supplicant (NAUN3 Device):** Initiates the authentication using client-side EAP-TLS and must be pre-provisioned with:

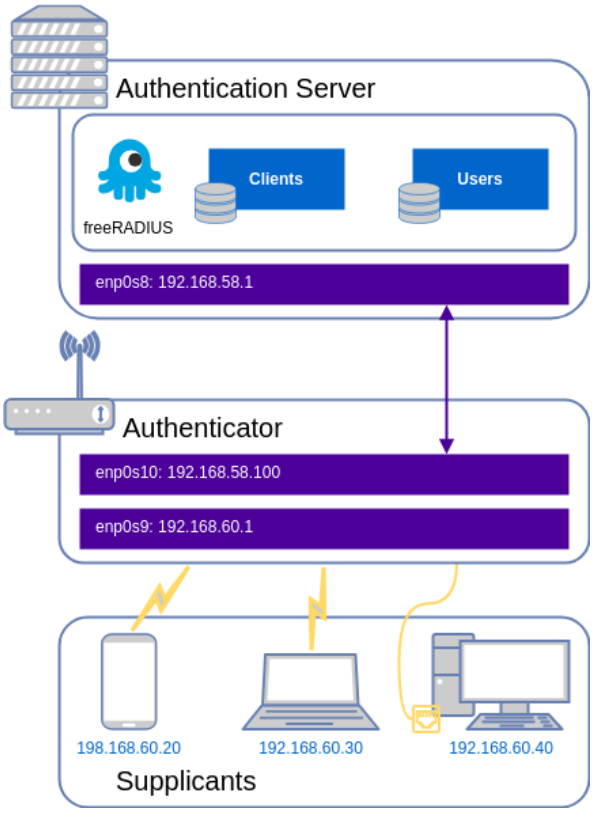


Figure 3.1: EAP-TLS Topology

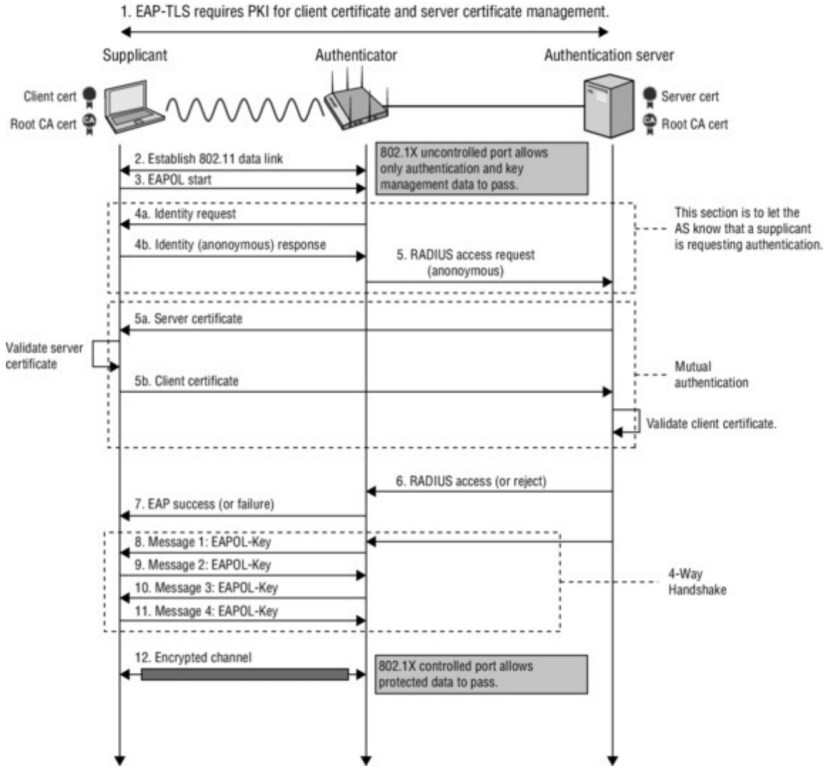


Figure 3.2: EAP-TLS Authentication Flow

- A unique identity (e.g., username or MAC address)
- A Certification Authority (CA) certificate to validate the server
- A client certificate and private key (optionally password-protected)

Standard software such as `wpa_supplicant` is used to handle this role.

2. **Authenticator (5G-RG):** Acts as a relay, not terminating the EAP-TLS session. Responsibilities include:
   - Detecting device connection attempts
   - Initiating the EAP exchange (e.g., via `hostapd`)
   - Relaying EAP messages over RADIUS through the `backhaul` DNN
   - Maintaining device authentication states
   - Triggering PDU Session establishment upon successful authentication
3. **Authentication Server (e.g., FreeRADIUS):** Typically managed by the same Internet Service Provider (ISP) operating the 5GC. Its functions are:
   - Managing identities and issuing credentials
   - Terminating and validating the EAP-TLS session
   - Authenticating clients via certificate verification
   - Returning EAP-Success or EAP-Failure via RADIUS

The EAP-TLS authentication procedure, shown in Figure 3.2, follows these steps:

1. Supplicant establishes a link-layer connection with the 5G-RG (e.g., via Wi-Fi or Ethernet).
2. It sends an EAPOL-Start to initiate authentication.
3. The 5G-RG replies with an EAP-Request/Identity.
4. The Supplicant responds with an EAP-Response/Identity (potentially anonymous).
5. The 5G-RG relays this via a RADIUS Access-Request to the Authentication Server.
6. The server replies with its certificate inside an EAP-Request (via Access-Challenge).
7. The Supplicant validates the server certificate using its CA.
8. The client certificate is returned via EAP-Response (also through Access-Challenge).
9. Mutual authentication is completed when the server verifies the client certificate.
10. The server sends RADIUS Access-Accept with EAP-Success (or Access-Reject with EAP-Failure).
11. The 5G-RG relays the final EAP result to the Supplicant.
12. If successful, a secure link-layer connection is established (e.g., Wi-Fi Protected Access (WPA)2 4-Way Handshake).

The 5G-RG interprets the final EAP message as follows:

- **EAP-Success:** Triggers the 5G-RG to initiate a PDU Session via the `clients` DNN, binding the session to the authenticated device and handling traffic mapping.
- **EAP-Failure:** Denies network access, terminating the process.

This mechanism assumes that devices are securely provisioned in advance with the required credentials. The provisioning process itself is considered out of scope for this implementation.

3.4 Proposed Identity Management Solution

Following the successful local authentication of a Wi-Fi-only/NAUN3 device, the next critical challenge is to enable its recognition and management within the 5GC.. This section details the proposed identity management solution, which innovatively utilizes the existing 5G PDU session framework to create a dynamic, per-device proxy identity. This approach allows the 5G-RG to mediate connectivity and map individual devices to distinct network sessions, thereby integrating them transparently into the 5G ecosystem.

3.4.1 The Identity Management Challenge

The deployment of Wi-Fi-only, or NAUN3, devices in the 5G network poses a fundamental identity management problem. Standard 5G identification depends on the SUPI, usually derived from credentials stored securely in a USIM (such as an IMSI or NAI). The SUPI over the air is encrypted as a SUCI. Devices without a USIM cannot create a SUPI or SUCI, and hence cannot be recognized, verified, or managed via standard 5GC processes.

3.4.2 Core Concept: PDU Sessions as Proxy Identities

To remedy this, the solution takes advantage of the capabilities of the 5G-RG and the versatility of 5G session management. From the viewpoint of the 5GC, a 5G-RG will act like a regular UE, including its own USIM, credentials, and support for several simultaneous PDU Sessions. This allows segregation of traffic for different services or endpoints.

Taking a cue from the idea of CGIDs, in which PDU Sessions might represent a set of devices behind a gateway (collectively by SSIDs or by Ethernet ports), the solution suggests a finer-grained approach: allocating each successfully authenticated NAUN3 device its very own dedicated PDU Session, and not aggregating them.

In the proposed model, every PDU Session, created and owned by the 5G-RG, will be a proxy identifier of a particular device. This enables not just integration into the 5GC, but also the ability to define session-specific controls and policies, and thus personalized management.

3.4.3 Establishing the Proxy Identity

After a device has been successfully authenticated through EAP-TLS, as described in the earlier section, the 5G-RG then initiates a PDU Session establishment procedure. This is done through its own credentials and SUPI, to the SMF, through the AMF, and specifying the special-purpose `clients` DNN.

Upon the assignment of resources, like an IP address, by the 5GC to the session, the session is bound by the 5G-RG to the authenticated NAUN3 device. From there on, the session becomes the device's operational identity in the 5G system.

3.4.4 Gateway's Role in Identity Mapping

The 5G-RG maintains an internal mapping table that links each NAUN3 device's local identifier (e.g., MAC address) to its associated PDU Session ID. This mapping enables accurate routing of data between the local network and the corresponding session within the 5GC.

3.4.5 5GC Perspective and Management

From the point of view of the 5GC, the procedure is all standard. It communicates to just one registered UE, the 5G-RG, to establish and release numerous PDU Sessions related to the `clients` DNN. Each session is managed by the SMF, and it follows traditional techniques for resource assignment, policy control, and life cycle management.

Specifically, the 5GC has no knowledge of the device identities or the authentication information of the end devices behind the 5G-RG, it merely controls the session at the request of the gateway.

3.4.6 Lifecycle of the Identity Mapping

The 5G-RG takes charge of the entire life-cycle of the proxy identity:

- **Creation:** Upon successful EAP-TLS authentication of an NAUN3 device, the gateway creates a mapping entry and asks the 5GC to create a dedicated.

- **Maintenance:** The 5G-RG routes the device's traffic through the corresponding PDU Session and monitors the device's local status (e.g., association state or heartbeat checks).
- **Termination:** If the device disconnects or becomes inactive:
  1. The gateway deauthenticates the device locally (e.g., via `hostapd`).
  2. It releases the associated PDU Session via a standard release procedure to the 5GC.
  3. The internal mapping entry is removed.

3.4.7 Advantages of the Proposed Approach

This session-based proxy identification scheme offers a number of beneficial features:

- **Transparency:** The solution is transparent to both the NAUN3 device (which experiences usual local verification) and the 5GC (which maintains regular PDU sessions).
- **Reuses Existing Infrastructure:** It constructs on top of existing 5G session management without the need to make any modifications to the core identity frameworks.
- **Localized Complexity:** Complexity of all sorts is centralized in the 5G-RG, reducing the need to integrate with.
- **Fine-Grained Control:** End-device PDU Sessions support per-device policy enforcement (i.e., traffic shaping or QoS) at the 5GC, providing fine-grained control.

3.5 Framework Architecture and Integration

3.5.1 Overall Architecture Overview

This top-level architecture Figure 3.3 shows the integration framework. The NAUN3 device connects locally (through Wi-Fi or cable) to the 5G-RG, the latter serving as a regular UE to the gNB and to the 5GC and serving as an EAP Authenticator forwarding requests to the external EAP-TLS Authentication Server within the domain of the ISP. The connection between the 5G-RG and the EAP server is done through the use of RADIUS, over a dedicated PDU session belonging to the `backhaul` DNN. Most importantly, when the local authentication of the 5G-RG is successful, the latter creates a different PDU session for the NAUN3 device.

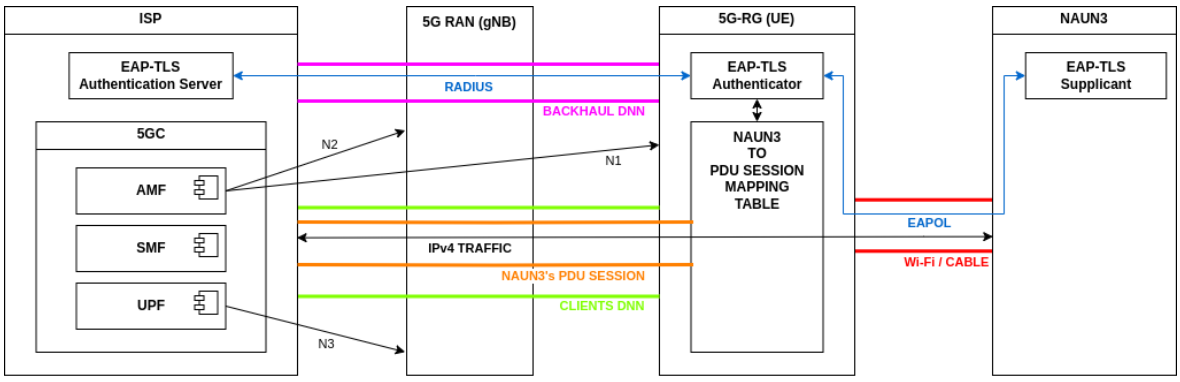


Figure 3.3: Overall Architecture

3.5.2 Interface and Protocol Integration

A key aspect of this framework's design is its reliance on standard, well-defined interfaces and protocols, minimizing the need for proprietary extensions. The integration is achieved by orchestrating these standard elements in a specific manner, primarily through the logic implemented within the 5G-RG.

The main interfaces and protocols involved are:

- **Local Network Interface (between NAUN3 Device and the 5G-RG):**
  - **Link Layer:** Standard Ethernet (Institute of Electrical and Electronics Engineers (IEEE) 802.3) or Wi-Fi (IEEE 802.11).
  - **Authentication:** Extensible Authentication Protocol over Local Area Network (LAN) (EAPOL - IEEE 802.1X) is used to transport EAP messages over the local link.
  - **EAP Method:** EAP-TLS is used for mutual authentication between the NAUN3 device (supplicant) and the EAP infrastructure (via the 5G-RG relay).
- **5G Interfaces (between 5G-RG and the 5GC):**
  - **N1 Interface:** Carries NAS signaling between the 5G-RG and the AMF for registration, authentication (of the RG itself), and session management procedures.
  - **N2 Interface:** Carries NGAP signaling between the gNB, which connects the 5G-RG, and the AMF, primarily for UE context management and PDU session resource setup requests related to the 5G-RG.
  - **N3 Interface:** Carries the user plane traffic encapsulated in General Packet Radio Service Tunnelling Protocol - User Plane (GTP-U) tunnels between the gNB and the UPF. This includes traffic for both the `backhaul` PDU session and all the individual `clients` PDU sessions.
- **Authentication Interface (between 5G-RG and the EAP Authentication Server):**
  - **Application Layer:** RADIUS protocol is used to carry EAP messages between the 5G-RG (acting as a RADIUS client and EAP authenticator) and the external authentication server).
  - **Transport:** RADIUS messages are transported over IP, using User Datagram Protocol (UDP). This IP traffic is securely tunneled through the 5G-RG's dedicated `backhaul` PDU session via the N3 interface and UPF.

The novelty lies not in modifying these protocols but in configuring the system components (5GC NFs, 5G-RG, EAP Server) and implementing the orchestration logic within the 5G-RG to manage the per-device proxy identity using standard 5G session management procedures.

3.6 Scope and Assumptions

This framework is able to bring NAUN3 devices into the 5G ecosystem by coordinating standard protocols and interfaces using smart logic centralized within the 5G-RG. Its main innovation is using individual PDU Sessions, set up by the 5G-RG, as flexible proxy identities

for each locally authenticated NAUN3 device. This method allows devices with no 5G support to still connect and interact natively with other 5G devices

Integration is handled dynamically across the device's connection lifecycle. After a device completes local EAP-TLS authentication, the 5G-RG requests a dedicated PDU Session (under the `clients` DNN) from the 5GC. The 5G-RG keeps an internal mapping between the device's local identity and its assigned PDU Session. If the device disconnects or loses connectivity, the 5G-RG promptly terminates the related PDU Session to free up 5G resources, ensuring only active and authenticated devices consume network capacity.

Lastly, this setup enables the 5GC to manage each device individually, applying policies like QoS and routing, through standard PDU Session management, all without requiring changes to the NAUN3 devices or 5G core network functions. Integration relies entirely on configurations and the gateway's mediation, avoiding any need to alter fundamental 5G protocols or the capabilities of the devices themselves.

CHAPTER 4

Development And Implementation

Building upon the proposed framework, this chapter describes the practical development and implementation process undertaken. In this thesis, the specific tools, technologies, and configurations used to realize the solution are described, including the construction of key modules and the configuration of the experimental environment required for subsequent validation.

4.1 Development Environment and Tools

To construct and validate the proposed framework, a virtualized multi-Virtual Machine (VM) environment was orchestrated using Vagrant with VirtualBox as the provider. This approach allowed for the creation of a reproducible and isolated network testbed. The environment consists of four distinct VMs, each running Ubuntu 22.04 LTS (Jammy Jellyfish) as the base operating system. The roles and typical resource allocations for these VMs, as defined in the `Vagrantfile`, are:

1. **`core` VM:** Hosts the 5GC functions and the EAP Authentication Server. Allocated 2Gigabyte (GB) Random Access Memory (RAM) and 1 Central Processing Unit (CPU).
2. **`gnb` VM:** Runs the gNB simulator. Allocated 1GB RAM and 1 CPU.
3. **`ue` VM:** Represents the 5G-RG, acting as a UE towards the 5GC and as an EAP Authenticator/Gateway towards the NAUN3 device. Allocated 1GB RAM and 1 CPU.
4. **`naun3` VM:** Simulates the Wi-Fi-only/NAUN3 end device, acting as an EAP Supplicant. Allocated 1GB RAM and 1 CPU.

The following core software components and tools were utilized across these VMs, installed and configured via shell scripts executed during Vagrant provisioning:

1. **5G Network Simulation:**
   - **Open5GS:** The open-source implementation of 5GC functions (AMF, SMF, UPF, NF Repository Function (NRF), AUSF, UDM, UDR, PCF, Network Slice Selection Function (NSSF)). Installed on the `core` VM.

- **MongoDB:** Used as the database backend for Open5GS, storing subscriber information and network function configurations. Installed from the official MongoDB repositories on the `core` VM.
- **UERANSIM:** An open-source gNB and UE simulator. Cloned from its GitHub repository and compiled from source on the `gnb` VM (for gNB functionality) and the `ue` VM (for UE/5G-RG functionality). The `nr-cli` utility from UERANSIM was also made available

2. **Authentication Infrastructure:**
   - **FreeRADIUS:** Employed as the EAP-TLS Authentication Server. Installed on the `core` VM and configured to handle EAP-TLS, manage client (UE/5G-RG) definitions, and generate/use X.509 certificates.
   - **`hostapd`:** Utilized as the EAP Authenticator on the `ue` VM (5G-RG). Cloned from its official repository (`w1.fi/hostap.git`) and compiled from source with the `CONFIG_DRIVER_WIRED=y` option enabled to support EAP over wired interfaces for the NAUN3 device connection.
   - **`wpa_supplicant`:** Used as the EAP Supplicant on the `naun3` VM. Installed via `apt` and configured to perform EAP-TLS authentication using client certificates.
3. **Networking and Utility Tools:**
   - **`dnsmasq`:** Configured as a Dynamic Host Configuration Protocol (DHCP) server on the `ue` VM to provide IP addresses to NAUN3 devices connecting to its local network interface (`enp0s9`).
   - **`yq`:** A command-line YAML processor, installed via `snap`. Extensively used in provisioning scripts to modify Open5GS and UERANSIM configuration files (e.g., setting IP addresses, DNNs, Access Point Names (APNs)).
   - **Build Tools:** `make`, `git`, `gcc`, `g++`, `cmake` (via `snap`), `libsctp-dev`, `lksctp-tools`, `pkgconf`, `libssl-dev`, `libnl-3-dev`, `libnl-genl-3-dev` were installed for compiling UERANSIM and `hostapd` from source.
   - **System Utilities:** `iproute2`, `net-tools`, `curl`, `gnupg` were used for network configuration and repository management.
   - **Node.js and Nginx:** Installed on the `core` VM to support and expose the Open5GS WebUI.
4. **Custom Tools and Scripts**
   - **`open5gs-dbctl`:** A shell script provided and used on the `core` VM to interact with the MongoDB database for managing Open5GS subscriber entries (adding UEs, defining APNs and slices).
   - **`interceptor`:** A custom Go application (compiled from source located in an `interceptor` directory, as indicated in the `Vagrantfile`) deployed on the `ue` VM. This is the key tool developed to orchestrate the logic for monitoring `hostapd` events and managing PDU sessions. It's specific internal workings are detailed later.

- **Provisioning Scripts:** A set of shell scripts (`core_install`, `gnb_install`, `ue_install`, `naun3_install`, `auth_server_install`, `ueransim_install`) were used by Vagrant to automate the installation and configuration of all software components on their respective VMs.

4.1.1 Network Topology and Configuration Management

The `Vagrantfile` defines several private networks (see Figure 4.1) to interconnect the VMs, establishing distinct network segments for communication between the 5GC and gNB (`192.168.56.0/24`), `gNB` and UE/5G-RG (`192.168.57.0/24`), and the UE/5G-RG's local network for NAUN3 devices (`192.168.60.0/24`). IP addresses for various interfaces and services (e.g., `CORE_IP`, `GNB_IP_CORE`, `UE_LAN_IP`, `AUTH_SERVER_IP` for RADIUS communication over the `backhaul` tunnel) are explicitly defined and passed as arguments to the provisioning scripts.

Vagrant's synced folder feature was utilized to share:

- EAP/RADIUS certificates generated by FreeRADIUS on the `core` VM to the `naun3` VM (via `/certs` on the guest).
- Runtime logs from all VMs to a `./build/runtime-logs` directory on the host machine.
- The compiled `interceptor` binary to the `ue` VM.

This comprehensive setup provides a fully functional, albeit simulated, environment for developing and testing the proposed solution for integrating NAUN3 devices into a 5G network.

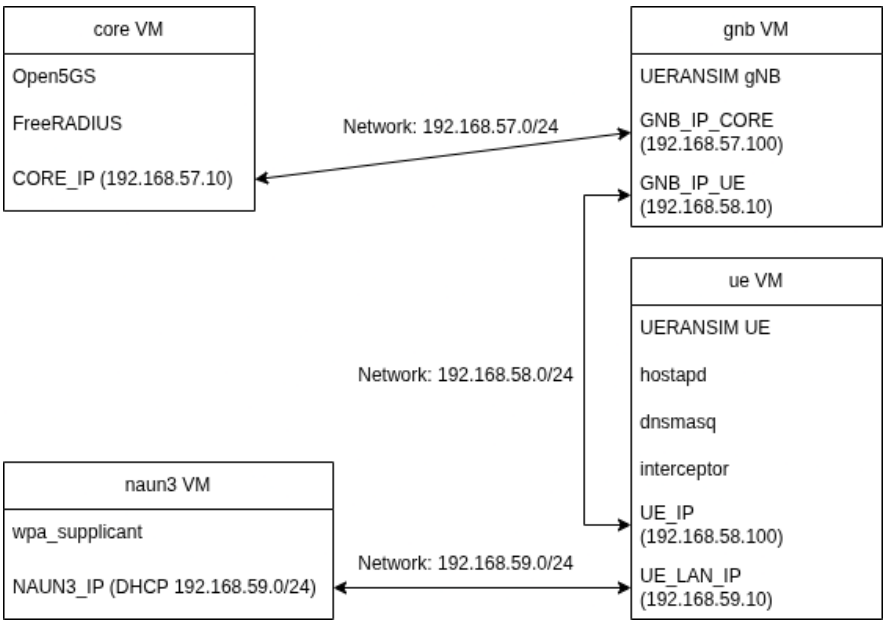


Figure 4.1: Vagrant deployed VMs with respecting services and interconnecting networks topology

4.2 Implementation of Proposed Authentication Logic

This section details the practical implementation of the EAP-TLS authentication mechanism, which forms the first crucial step in integrating NAUN3 devices. The setup involves configuring an EAP Authentication Server (FreeRADIUS), an Authenticator (`hostapd` on the 5G-RG), and an EAP Supplicant (`wpa_supplicant` on the NAUN3 device), with a custom Go application (`interceptor`) orchestrating the events on the 5G-RG.

Firstly, the EAP Authentication Server was implemented using FreeRADIUS on the `core` VM, provisioned by the `auth_server_install` script. This script is responsible for configuring FreeRADIUS and the keys it uses.

Regarding RADIUS Client Configuration, the 5G-RG (`ue` VM) was registered as a RADIUS client in `/etc/freeradius/3.0/clients.conf`. This entry specified the `ue` VM's IP address designated for EAP traffic (the `CLIENT_EAP_IP` variable defined in the `Vagrantfile` as `10.45.0.2`) and a shared secret (the `CLIENT_SECRET` variable also defined in the `Vagrantfile`, holding a randomly generated value) for securing RADIUS communication.

FreeRADIUS provides the mechanisms necessary to generate a Public Key Infrastructure (PKI) via a script which generates the following:

- A CA certificate (`ca.pem`)
- A server certificate (`server.pem`) and private key (`server.key`) for FreeRADIUS itself.
- A client certificate (`client.p12`, a Public-Key Cryptography Standards 12 (PKCS12) bundle containing the certificate and private key) for the NAUN3 device. Passwords for these certificates (`CERT_CA_PASSWD`, `CERT_SERVER_PASSWD`, `CERT_CLIENT_PASSWD`) are randomly generated value in the `Vagrantfile` and passed to the script.

Also, the EAP module in FreeRADIUS (`/etc/freeradius/3.0/mods-available/eap`) was to modified such as:

- To set the EAP type to Transport Layer Security (TLS) - `default_eap_type = tls`.
- The server's private key password is set - `private_key_password`
- The paths for the certificates and private keys are set - `private_key_file`, `certificate_file` and `ca_file`

The necessary certificates for the NAUN3 device (CA certificate `ca.pem` and the client PKCS12 bundle `client.p12`) were copied from the `core` VM's FreeRADIUS certificate directory to a Vagrant synced folder, making them accessible to the `naun3` VM.

Secondly, the 5G-RG (`ue` VM) acts as the EAP Authenticator, using `hostapd`. Its setup is managed by the `ue_install` script.

Starting installation, `hostapd` is cloned from `w1.fi/hostap.git` and compiled from source, ensuring the `CONFIG_DRIVER_WIRED=y` option was enabled in its `.config` file to support EAP authentication over the virtual interface connecting both the `ue` VM to the `naun3` VM.

Regarding the `hostadp` configuration, which is the entity responsible for LAN authentication, we can review it bellow (see Listing 4.1).

```
interface=enp0s9
driver=wired
ctrl_interface=/var/run/hostapd
ctrl_interface_group=vagrant

logger_syslog=-1
logger_syslog_level=0

ieee8021x=1
own_ip_addr=$CLIENT_EAP_IP

auth_server_addr=$AUTH_SERVER_IP
auth_server_port=1812
auth_server_shared_secret=$CLIENT_SECRET
```

Listing 4.1: `hostapd` configurations

- `auth_server_addr`: Set to `AUTH_SERVER_IP` (the IP of the `core` VM (`10.45.0.1`) reachable via the `backhaul` PDU session).
- `auth_server_port=1812`.
- `auth_server_shared_secret`: Set to `CLIENT_SECRET`.
- `own_ip_addr`: Set to `CLIENT_EAP_IP` (`10.45.0.2`), ensuring RADIUS packets from `hostapd` originate from the correct IP address within the `backhaul` PDU session

When implementing the `naun3` device, it configured to act as an EAP Supplicant using `wpa_supplicant`, as detailed in the `naun3_install` installation script.

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=vagrant
eapol_version=2
ap_scan=0

network={
    eapol_flags=0

    key_mgmt=IEEE8021X
    eap=TLS

    identity="user@example.org"
    ca_cert="/certs/ca.pem"
    private_key="/certs/client.p12"
    private_key_passwd="$CERT_CLIENT_PASSWD"
}
```

Listing 4.2: `wpa_supplicant` configurations

In the configurations shown in the Listing 4.2, a network block was defined specifically for EAP-TLS authentication:

- `key_mgmt=IEEE8021X`.
- `eap=TLS`.
- `identity="user@example.org"` was used as the EAP identity.

- `ca_cert="/certs/ca.pem"` specified the path to the CA certificate (obtained via the synced folder).
- `private_key="/certs/client.p12"` specified the path to the client's PKCS12 certificate/key bundle.
- `private_key_passwd` was set to `CERT_CLIENT_PASSWD`.

Also, `wpa_supplicant` was configured to use the wired driver (via the `-Dwired` argument) for its network interface (via the `-ienp0s8` argument). The scan for Access Points (APs) is also disabled via `ap_scan=0`, which is appropriate for wired connections.

When it comes to orchestrating the authentication events, the custom Go application, `interceptor.go`, deployed on the `ue` VM, plays a pivotal role in bridging the EAP authentication outcome with the 5G session management logic.

In it's interaction with `hostapd`, the `interceptor` establishes a connection to `hostapd`'s control interface socket (e.g., `/var/run/hostapd/enp0s9`) using a Unix domain socket. It sends an `"ATTACH"` command to subscribe to `hostapd` events, and the `HostapdListener` goroutine continuously monitors messages from `hostapd`.

To detect EAP successes, it specifically parses incoming messages for the string `"CTRL-EVENT-EAP-SUCCESS"`. An when one is detected, the `interceptor` extracts the MAC address of the authenticated NAUN3 device from the event message (e.g., `CTRL-EVENT-EAP-SUCCESS aa:bb:cc:dd:ee:ff`). This serves as a primary trigger for the `interceptor` that then proceeds to manage the device in its `allowed_devices` map and initiates the logic for establishing a new PDU session for this device.

4.2.1 Implemented EAP-TLS Authentication Flow Summary

The implemented authentication sequence is as follows:

1. The `naun3` VM (`wpa_supplicant`) attempts to authenticate over its `enp0s8` interface connected to the `ue` VM's (5G-RG) `enp0s9` interface.
2. `hostapd` on the `ue` VM detects the EAPOL-Start and initiates the EAP-TLS exchange.
3. EAP-TLS messages are relayed by `hostapd` to the FreeRADIUS server on the `core` VM. This communication occurs via RADIUS packets, which are transported over the 5G-RG's `backhaul` PDU session (using `CLIENT_EAP_IP` as the source and `AUTH_SERVER_IP` as the destination).
4. FreeRADIUS validates the NAUN3 device's client certificate against its CA and configuration.
5. Upon successful validation, FreeRADIUS sends a RADIUS Access-Accept containing an EAP-Success message back to `hostapd`.
6. `hostapd` relays the EAP-Success to `wpa_supplicant` on the NAUN3 device and concurrently emits the `CTRL-EVENT-EAP-SUCCESS <NAUN3_MAC_ADDRESS>` message to its control interface.
7. The `interceptor` application on the `ue` VM captures this event, confirming the NAUN3 device's successful local authentication and readiness for the next stage of network integration.

This setup effectively implements the EAP-TLS authentication flow, using standard tools configured to interact in a specific way, with the custom interceptor acting as the crucial link to the 5G-specific actions.

4.3 Implementation of Identity Management Mechanisms

With local EAP-TLS authentication successfully implemented, this section details how the framework manages a unique network presence for each authenticated NAUN3 device. Given that these devices lack native 5G identifiers (SUPI/SUCI), the core of the implemented solution is the dynamic establishment and management of a dedicated PDU Session by the 5G-RG for each NAUN3 device. This PDU Session effectively serves as its proxy identity within the 5G network. The custom Go application, running on the 5G-RG (`ue` VM), is the central orchestrator of this mechanism.

4.3.1 1. Triggering Proxy Identity Establishment

The process of establishing a proxy identity for an NAUN3 device is initiated immediately after its successful local authentication. The `HostapdListener` *goroutine* (defined in `hostapd_interceptor.go` and launched by `main.go` monitors `hostapd`'s control interface. Upon detecting the `CTRL-EVENT-EAP-SUCCESS <MAC_ADDRESS>` message (constant `hostapdEventEAPSuccess`), the listener extracts the MAC address of the successfully authenticated NAUN3 device. This event and the device's MAC address serve as the trigger for the subsequent identity management steps.

4.3.2 2. PDU Session Creation by the Orchestration Logic

Once an NAUN3 device is authenticated, the `HostapdListener` calls the `NewPDUSession` function (from `ueransim_pdu_handler.go`. This function is responsible for requesting a new PDU session from the 5GC via the 5G-RG's UE stack (UERANSIM).

- It constructs and executes the UERANSIM command-line interface tool: `nr-cli <5G-RG_IMSI> -exec "ps-establish IPv4 -sst 1 -dnn <DNN_NAME>"`.
  - The `<5G-RG_IMSI>` (e.g., `999700000000001`) is the pre-configured IMSI of the 5G-RG itself, passed as a command-line argument (`-imsi`) to the main application and subsequently to the `HostapdListener` and `NewPDUSession`.
  - The `<DNN_NAME>` (e.g., `clients`) is also passed as a command-line argument (`-dnn`) and used to target the PDU session request. This DNN was specifically configured in Open5GS on the `core` VM and made known to the UERANSIM UE stack on the `ue` VM.
- After requesting the session, `NewPDUSession` enters a polling loop, repeatedly calling `LastPDUSession` (which executes `nr-cli <5G-RG_IMSI> -exec "ps-list"` and parses its YAML output) until the newly requested session transitions to the `"PS-ACTIVE"` state (`pduSessionStateActive`) and has an IP address assigned by the 5GC. A timeout mechanism with retries (`pduSessionEstablishRetries`, `pduSessionEstablishInterval`) is implemented.

4.3.3 3. Gateway-Managed Internal Mapping

The main application maintains a global map: `allowedDevices` which is a key-value map, that identifies `Device` objects using their MAC addresses as keys. This `Device` object (defined in `network_handler.go`) stores the state.

- When an NAUN3 device successfully authenticates and its dedicated PDU session (type `Session` from `ueransim_pdu_handler.go`) becomes active, an entry is added to this map by the `HostapdListener`. The MAC address of the NAUN3 device serves as the key.
- The `Device` object stores crucial information: its current `state` (e.g., `"AUTHENTICATED"`, `"LEASED"`, `"REACHABLE"`), a pointer to the `Session` object (containing PDU Session ID, state, APN/DNN, and the 5GC-allocated IP Address), Lease information (from `dnsmasq_handler.go`, and `AppliedIPTablesRules` (a slice of `AppliedRuleDetail` from `routing_handler.go`. This map is central to linking the local device to its 5G network representation and its specific traffic routing rules.

4.3.4 4. NAUN3 Device Local IP Addressing

For the NAUN3 device to communicate on the local network segment:

- Upon successful PDU session establishment, the `HostapdListener` calls `AllowMAC()` (from `dnsmasq_handler.go`). This function appends `dhcp-host=<NAUN3_MAC_ADDRESS>,<LEASE_TIME>,set:known` to `/etc/allowed-macs.conf` on the 5G-RG (`ue` VM). The `leaseTime` is passed as a command-line argument (`-lease-time`) to `main.go`.
- `dnsmasq` on the 5G-RG (configured in `ue_install`) serves DHCP only to MAC addresses listed as `"known"` in this file.
- `wpa_supplicant` on the NAUN3 VM then executes `sudo dhclient enp0s8` to obtain a local IP address (e.g., from `192.168.60.0/24`).

4.3.5 5. Proxy Identity Lifecycle Management (Termination)

The `HostDisconnectListener` *goroutine* (in `network_handler.go`, launched by `main.go`) and the `ForgetDevice` function (in `network_handler.go`) manage the termination of the proxy identity.

- `HostDisconnectListener` periodically checks device reachability on the LAN interface (e.g., `enp0s9`, derived from the `-interface` flag passed to `main.go`) using `netlink.NeighList` (which is based on the `ip neighbour` too). A device is considered for removal if its state becomes stale/failed and its DHCP lease is significantly into its expiry, or if the MAC is no longer in the Address Resolution Protocol (ARP)/neighbor list.
- This triggers the `ForgetDevice` function, which performs cleanup:
  - `DisallowMAC()` (from `dnsmasq_handler.go`): Removes the NAUN3 device's MAC address entry from the `allowedMACsFilePath` and the `leasesFilePath` (passed

via `-leases` flag to `main.go`), followed by a `RestartDnsmasq()` call which will force `dnsmasq` to restart and thus reload the configurations now updated.
  - `Deauth()` (from `hostapd_interceptor.go`): Sends a `"DEAUTHENTICATE <MAC_ADDRESS>"` command to `hostapd`.
  - `ruleManager.RemoveRulesForDevice()` (from `routing_handler.go`): Removes the specific `iptables` rules and `ip route/rule` entries that were previously applied for this device, using the `AppliedIPTablesRules` stored in the `Device` object. The `ruleManager` is initialized in `main.go`.
  - `ReleasePDUSession()` (from `ueransim_pdu_handler.go`): Executes `nr-cli <5G-RG_IMSI> -exec "ps-release <PDU_SESSION_ID>"` to terminate the dedicated clients PDU session associated with this NAUN3 device.
  - Finally, the NAUN3 device's entry is removed from the global `allowedDevices` map.
- The `DnsmasqListener` (in `dnsmasq_handler.go`, launched by `main.go`) also monitors the DHCP lease file (`leasesFilePath`), updating lease details (`expiration`, `counter`) and transitioning the device state to `"LEASED"` within the `allowedDevices` map upon lease acquisition/renewal.

4.3.6 6. Implemented Traffic Mapping

The `interceptor` system, through the `routing_handler.go` module, implements concrete traffic mapping for each authenticated NAUN3 device (as shown in Figure 4.2), ensuring its traffic is routed via its dedicated PDU session.

- After a PDU session is successfully established for an NAUN3 device, the `HostapdListener` calls `ruleManager.ApplyMappingRules()`. The `ruleManager` instance is initialized in `main.go` via `NewRuleManager()`.
- `ApplyMappingRules` is passed the LAN interface name (e.g., `enp0s9`, from `-lan-if` flag), the NAUN3's MAC address, the PDU session's tunnel interface name (e.g., `uesimtun<ID>`, where `ID` is the PDU session ID), the PDU session's gateway IP (e.g., `10.46.0.1` for the `clients` DNN, from `-pdu-gw-ip` flag), and the PDU session `ID`.
- The function then systematically configures policy-based routing and Network Address Translation (NAT) using `iptables` and `ip` commands:
  1. **Custom Routing Table:** An entry for a new routing table (e.g., `200+<PDU_ID> table_pdu_<PDU_ID>`) is added to `/etc/iproute2/rt_tables` using the `manageRTTableEntry` function. This ensures the table definition persists.
  2. **Default Route in Custom Table:** A default route `ip route add default via <PDU_GATEWAY_IP> dev <PDU_IF_NAME> table table_pdu_<PDU_ID>` is added, directing all traffic for this custom table out through the NAUN3's specific PDU session interface.
  3. **Policy Rule:** An IP policy rule `ip rule add fwmark <PDU_ID> table table_pdu_<PDU_ID>` is created. This rule directs any packet marked with the PDU session ID (as a firewall mark) to use the newly created custom routing table.

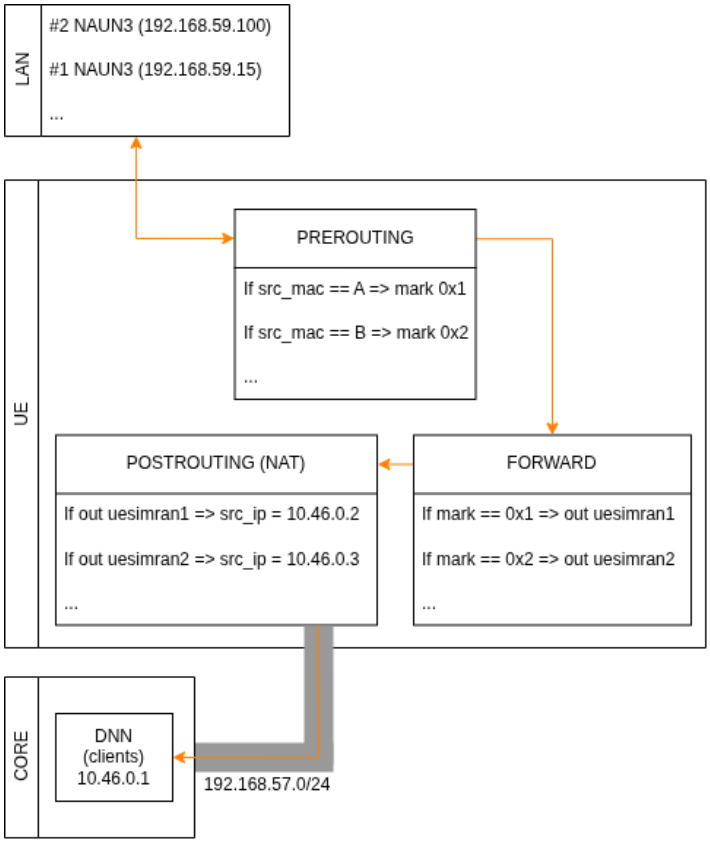


Figure 4.2: Policy Based Routing

4. **Packet Marking:** An `iptables` rule is inserted into the `mangle` table's `PREROUTING` chain: `-i <LAN_IF> -m mac -mac-source <NAUN3_MAC> -j MARK -set-mark <PDU_ID>`. This rule marks all incoming packets from the specific NAUN3 device's MAC address on the LAN interface with its corresponding PDU session ID.
5. **Forwarding:** An `iptables` rule in the `filter` table's `FORWARD` chain explicitly allows marked packets originating from the NAUN3's MAC on the LAN interface to be forwarded to its designated PDU session interface: `-i <LAN_IF> -o <PDU_IF_NAME> -m mac -mac-source <NAUN3_MAC> -m mark -mark <PDU_ID> -j ACCEPT`. A global `FORWARD` policy of `DROP` is assumed or set by `NewRuleManager`.
6. **NAT (Masquerade):** An `iptables` rule in the `nat` table's `POSTROUTING` chain: `-o <PDU_IF_NAME> -j MASQUERADE` performs Source Network Address Translation (SNAT) for all traffic exiting via the PDU session interface, making it appear to originate from the IP address assigned to that PDU session.

- These dynamically applied rules ensure that traffic from each authenticated NAUN3 device is uniquely marked, routed through its dedicated PDU session, and correctly *NATted* for external communication. The details of these applied rules are stored in the `Device.AppliedIPTablesRules` slice and are systematically removed by

`ruleManager.RemoveRulesForDevice()` during the `ForgetDevice` process to ensure a clean state.

4.4 Adaptation of Network Functions

The successful implementation of the proposed framework did not require code-level modifications to the standard 5G NFs or RAN/UE simulators. Instead, these components were adapted through specific configurations to support the gateway-centric authentication and identity management scheme for NAUN3 devices. The primary tools used for this were Open5GS for the 5GC, UERANSIM for the gNB and the 5G-RG's UE stack, all provisioned and configured via scripts within a Vagrant-managed virtualized environment.

4.4.1 Open5GS (5GC) on `core` VM

The Open5GS components whose configurations were specifically adapted for this framework were the AMF, SMF, and UPF. The AUSF and UDM operated in a standard manner for the 5G-RG's own registration and authentication. The relevant configurations were applied as follows.

- **Connectivity Configuration:** The AMF's NGAP interface and the UPF's GTP-U interface were bound to a designated IP address on the `core` VM, facilitating connectivity with the gNB over their respective private network segment. This was achieved by modifying `amf.yaml` and `upf.yaml` configuration files using `yq`.
- **DNN Configuration:** Two distinct DNNs were defined in the SMF and UPF configurations to segregate traffic:
  - `backhaul` DNN: Configured in `smf.yaml` and `upf.yaml`, associated with the default `ogstun` tunnel interface. This DNN provides the primary PDU session for the 5G-RG itself, used for operational traffic such as RADIUS communication with the EAP Authentication Server. The SMF configuration for this DNN assigns IPs from a dedicated subnet; the 5G-RG's UE stack is configured to use an IP from this subnet for EAP client communication, and the EAP server on the `core` VM is also reachable via an IP in this same `backhaul` network space.
  - `clients` DNN: Also configured in `smf.yaml` and `upf.yaml`. A dedicated tunnel interface (`clientun0`) was created on the `core` VM and assigned an IP address to serve a distinct subnet. The `clients` DNN was associated with this interface and subnet, enabling the SMF/UPF to allocate IPs from this specific range for the individual PDU sessions established for each NAUN3 device.
- **Subscriber Provisioning (for the 5G-RG):** The 5G-RG itself was provisioned as a standard UE in the Open5GS MongoDB database using the `open5gs-dbctl` script. This involved adding an entry with the 5G-RG's IMSI, pre-shared key, and Operator Key (OPC). This subscription was configured to allow the 5G-RG to establish PDU sessions on both the `backhaul` and `clients` DNNs using the `update_apn` command of `open5gs-dbctl.sh`.

4.4.2 UERANSIM (gNB on `gnb` VM and UE stack on `ue` VM)

The UERANSIM components were configured without any code modifications, using their standard YAML configuration files manipulated by `yq`:

- **gNB Configuration (`gnb` VM):** The `open5gs-gnb.yaml` configuration file was modified to set:
  - The gNB's link IP address for communication towards the UE (5G-RG), within their shared private network.
  - The gNB's NGAP and General Packet Radio Service Tunneling Protocol (GTP) IP addresses for N2 and N3 interface communication with the AMF and UPF respectively, within their shared private network.
  - The AMF's IP address for establishing the N2 connection.
- **5G-RG UE Stack Configuration (`ue` VM):** The `open5gs-ue.yaml` file for the UERANSIM instance representing the 5G-RG was configured to:
  - Specify the gNB's IP address in its search list.
  - Use the provisioned USIM credentials (IMSI, Key, OPC) identical to those configured in the Open5GS database.
  - Support multiple PDU sessions. The `ue_install.sh` script explicitly configures the first session (index 0) to use the `backhaul` APN/DNN. It then prepares subsequent session configurations (though `DEFAULT_PDU_SESSIONS` is set to `0` in the `Vagrantfile`, the script structure allows for more, and the interceptor dynamically requests them) to support Internet Protocol Version 4 (IPv4) type PDU sessions on the clients APN/DNN with Service Selection Tunneling (SST) `1`. This configuration enables the 5G-RG's UE stack to request the necessary PDU sessions as orchestrated by the custom interceptor application.

These configurations ensured that the standard 5G components could support the project's architecture, where the 5G-RG acts as a legitimate UE capable of managing multiple distinct data paths (via PDU Sessions on different DNNs) for its own operational needs and for proxying connectivity for the locally authenticated NAUN3 devices.

4.5 System Integration and Configuration

The various implemented components were integrated into a cohesive test environment using Vagrant for VM orchestration and shell scripts for provisioning. This ensured consistent configurations and network connectivity across the simulated 5G system and attached devices.

4.5.1 VM Orchestration and Network Topology:

The `Vagrantfile` defines four VMs (`core`, `gnb`, `ue`, `naun3`), each running Ubuntu 22.04 LTS. Vagrant establishes several private networks, each serving a distinct functional purpose, to facilitate communication:

- **5GC-gNB Network:** A dedicated private network connects the `core` VM (hosting Open5GS NFs) with the `gnb` VM. Specific IP addresses are assigned to each VM on this network for N2 (AMF-gNB) and N3 (UPF-gNB) interface traffic.
- **gNB-UE (5G-RG) Network:** Another private network connects the `gnb` VM with the `ue` VM (representing the 5G-RG). IPs on this network facilitate the simulated the radio interface.
- **5G-RG LAN:** A separate private network serves as the LAN for the `ue` VM, connecting its designated LAN interface to the `naun3` VM. This segment handles local EAPOL for authentication and the NAUN3 device's data traffic before it's routed into a PDU session.

  Vagrant's synced folders facilitate sharing of EAP/RADIUS certificates (from `core` to `naun3` via `/certs`), runtime logs from all VMs to the host (`./build/runtime-logs`), and the compiled `interceptor` binary to the `ue` VM (`./build/interceptor` to `/home/vagrant/interceptor`).

4.5.2 5G-RG (`ue` VM) as the Central Integration Hub:

The `ue` VM is the cornerstone of the integration, performing multiple functions simultaneously:

- **RAN Connectivity:** Its UERANSIM UE stack connects to the simulated gNB on the `gnb` VM.
- **Local Network Services:**
  - `hostapd` is configured on its LAN interface to provide 802.1X/EAP-TLS authentication for devices on its local network, relaying authentication requests to the FreeRADIUS server.
  - `dnsmasq` acts as a DHCP server for this LAN, dynamically assigning local IP addresses to authenticated NAUN3 devices based on the `/etc/allowed-macs.conf` file managed by the interceptor application.
- **`backhaul` PDU Session:** Upon registration with Open5GS, the 5G-RG's UE stack establishes its primary PDU session on the `backhaul` DNN. This session receives an IP address from the 5GC's `backhaul` subnet. This IP is then used as the source for RADIUS messages sent from `hostapd` (via the 5G-RG) to the FreeRADIUS server, which is also reachable within this `backhaul` network space.
- **`clients` PDU Sessions:** Orchestrated by the `interceptor` application, for each successfully EAP-authenticated NAUN3 device, the 5G-RG's UE stack requests a new, dedicated PDU session on the `clients` DNN. These sessions are assigned IPs from the distinct `clients` subnet by the SMF/UPF.
- **IP Forwarding and Traffic Routing:**
  - IP forwarding is enabled on the `ue` VM (`net.ipv4.ip_forward=1`).
  - The `interceptor` application, through its routing handler module, dynamically configures policy-based routing using `ip rule` and `ip route` commands, and `iptables` rules for packet marking (`mangle` table), NAT (`nat` table, `MASQUERADE`),

and forwarding (`filter` table). This ensures that traffic originating from an NAUN3 device on the local LAN is marked, routed through its dedicated `clients` PDU session tunnel interface (e.g., `uesimtunX`), and *NATted* with that PDU session's assigned 5GC IP address.

4.5.3 NAUN3 Device (`naun3` VM) Configuration:

The `naun3` VM connects to the 5G-RG's LAN interface. Its `wpa_supplicant` service is configured for EAP-TLS authentication using the client certificate and CA certificate shared via the Vagrant synced folder (`/certs`). Upon successful authentication, it obtains a local IP address via DHCP from the `dnsmasq` server on the 5G-RG.

4.5.4 EAP Authentication Server (FreeRADIUS on `core` VM)

FreeRADIUS on the `core` VM is configured to listen for RADIUS requests. The `ue` VM (5G-RG) is defined as a RADIUS client, identified by its `backhaul` PDU session IP address. It authenticates NAUN3 devices based on the client certificates issued by its internal CA.

4.5.5 Parameter Consistency

The `Vagrantfile` serves as the central point for defining critical network parameters (functional IP addresses, IMSI, keys, shared secrets, certificate passwords). These parameters are passed as arguments to the respective provisioning scripts (`*.sh`), ensuring consistency across the configurations of Open5GS, UERANSIM, FreeRADIUS, `hostapd`, and the command-line flags for the `interceptor` application. This integrated setup allows for the end-to-end simulation and testing of the proposed NAUN3 device integration framework.

4.6 Implementation Challenges

The development and implementation of this framework, while ultimately successful in the simulated environment, encountered several technical challenges. These ranged from complexities in orchestrating the virtualized 5G system to specific issues encountered when attempting to integrate physical hardware.

4.6.1 Orchestration of Simulated 5G Components

Regarding configuration complexity, setting up a multi-VM environment with Open5GS, UERANSIM, FreeRADIUS, `hostapd`, and `dnsmasq` required careful management of numerous configuration files and network parameters. Ensuring IP address consistency, correct DNN definitions, subscriber provisioning, and proper inter-component communication (e.g., RADIUS, NGAP, GTP-U) across different virtual networks demanded meticulous scripting (as seen in the Vagrant provisioning scripts). Any misconfiguration in one component often had cascading effects, making debugging a time-consuming process.

Also service dependencies and startup order need careful handling, ensuring that services started in the correct order and that dependencies were met (e.g., MongoDB before and Open5GS NFs before UERANSIM components could connect) was crucial and required careful scripting within the Vagrant provisioning process.

For dynamic PDU Session management with UERANSIM, while the `nr-cli` tool provides a command-line interface to manage PDU sessions, programmatically triggering and monitoring these from an external application (the custom orchestration logic) involved parsing command output and implementing polling mechanisms, which is less robust than a direct Application Programming Interface (API)-based interaction might be.

4.6.2 Development of the Custom Orchestration Logic

Handling the events and managing the state, the custom orchestration application needed to reliably capture events from `hostapd` (EAP success), manage the state of multiple NAUN3 devices (authentication status, associated PDU session, applied routing rules), and react to network events (DHCP lease changes, device disconnections via ARP/neighbor cache monitoring). Coordinating these asynchronous events and maintaining a consistent internal state for each device was a key challenge.

The custom logic for interfacing with system utilities, interacts with tools like `nr-cli` (for PDU sessions), `iptables`, and `ip route`/`ip rule` (for traffic mapping). Ensuring these commands were executed correctly with the appropriate parameters for each device, and handling their output or potential errors, required careful implementation and robust error checking within the Go application.

The developed `interceptor` tool is highly concurrency driven, managing multiple *goroutines* for listening to `hostapd`, `dnsmasq` leases, and network disconnects, while ensuring thread-safe access to shared data structures (like the map of allowed devices), required careful use of synchronization primitives.

4.6.3 Challenges with Physical Modem Integration (Quectel RG500Q-GL RedCap Attempt)

An attempt was made to integrate a physical 5G modem, specifically a Quectel RG500Q-GL (RedCap) Universal Serial Bus (USB) modem, to explore the feasibility of the solution with real hardware acting as the 5G-RG's connection to the 5G network. This presented significant challenges distinct from the simulated UERANSIM environment:

- **Proprietary Drivers and Kernel Dependencies:** The Quectel RG500Q-GL, being an experimental sample, relied on proprietary drivers provided by Quectel rather than standard Linux kernel drivers. These drivers had to be compiled from source and were highly sensitive to specific kernel versions. This severely restricted the choice of host operating system and often necessitated the use of a dedicated Single Board Computer (SBC) that met the kernel requirements, complicating the development workflow (requiring Secure Shell (SSH) access to the SBC for modem interaction).
- **Lack of Public Documentation:** Comprehensive public documentation for the modem's Attention (AT) commands, Qualcomm MSM Interface (QMI) interface (`qmicli`), and particularly for advanced features like establishing multiple concurrent PDU sessions with Qualcomm Multiplexing and Aggregation Protocol (QMAP) mode, was scarce or non-existent. This made configuring the modem for the project's specific needs (e.g.,

one `backhaul` PDU session, multiple `clients` PDU sessions) a process of trial, error, and reliance on limited provided snippets.

- **Difficulties with Multiple PDU Sessions:**
  - While AT commands and `qmicli` could be used to define Packet Data Protocol (PDP) contexts for different APNs/DNNs (e.g., `AT+CGDCONT` or `qmicli -wds-create-profile`), activating and managing multiple *simultaneous* PDU sessions, especially binding them to distinct virtual network interfaces for independent routing by the custom orchestration logic, proved extremely challenging with the provided Quectel tools (`quectel-qmi-proxy`, `quectel-CM`).
  - The available examples and tools from Quectel primarily demonstrated setting up multiple connections to different APNs but did not clearly address the scenario of multiple active PDU sessions to the *same* APN (our `clients` DNN) or robustly exposing these as distinct network interfaces to the Linux system in a way that the custom routing logic could easily manage.
  - The `qmicli` tool, while powerful, did not offer a straightforward or well-documented method for QMAP-based multiplexing of multiple PDU sessions that was confirmed to work with this specific modem model and firmware. Multiple attempts were made by trying variations of existing configuration steps for previous modem model's versions, but none were successful.
- **Contrast with UERANSIM:** The UERANSIM environment, by comparison, allowed for relatively straightforward programmatic control over PDU session establishment and release via `nr-cli`, making it a more tractable platform for developing and testing the core logic of the custom orchestration application and the overall framework. The complexities of the physical modem's driver and proprietary connection manager abstracted away much of the direct control needed for fine-grained, multi-session management.

These challenges highlight the gap that can exist between simulated environments and the specificity of physical hardware, especially when dealing with proprietary drivers and limited documentation for specialized or pre-release components. While the core concepts of the project were validated in simulation, porting to such physical hardware would require significant additional effort in driver-level integration and modem-specific control.

4.7 Latest 3GPP Developments

During the course of this project's development and implementation, 3GPP Release 19 specifications have evolved, introducing mechanisms that are highly pertinent to the challenge of managing non-3GPP devices behind a 5G-RG [33]. These developments, validate the problem space addressed by this thesis and underscore the industry's movement towards more granular control and differentiated services for such devices. This section discusses the relevance of these advancements to the implemented framework.

4.7.1 Synergies with the Implemented Approach

The implemented solution, which focuses on providing each NAUN3 device with a dedicated PDU Session triggered by local EAP-TLS authentication, finds resonance with several directions emerging in Release 19:

- **Individual Device Focus**: The formalization of the "Non-3GPP Device Identifier" [43] and procedures for mapping traffic from these identified devices to distinct QoS Flows [41] clearly signal the need for individualized treatment. The implemented framework achieves this by assigning a unique PDU Session, which inherently allows for individual policy and QoS application.
- **5G-RG as a Key Enabler**: 3GPP positions the 5G-RG as the entity responsible for local device traffic recognition and initiating procedures for differentiated QoS. The implemented `interceptor` service on the 5G-RG, along with configured `hostapd` and UERANSIM (acting as the UE stack for the RG), directly embodies this enhanced gateway role by managing local authentication and orchestrating 5G session management per NAUN3 device.
- **Dynamic Service Adaptation**: The 3GPP framework allows for dynamic QoS adjustments, potentially leading to PDU Session Modification or even new PDU Session Establishment for a device requiring differentiated service. The project's implementation of dynamic PDU session creation via the interceptor service upon successful authentication, and termination upon device departure, aligns with this principle of responsive service provisioning.
- **PDU Sessions for Traffic Segregation**: The use of PDU Sessions to segregate traffic for "Connectivity Group IDs" is an established 3GPP concept. The implemented solution extends this to its most granular form by dedicating a PDU Session to each NAUN3 device, thereby utilizing a standard 5G mechanism for achieving fine-grained traffic management and isolation.

4.7.2 Addressing the Authentication Gap in the Implemented Solution

A significant aspect where the implemented solution provides a specific contribution, in the context of 3GPP Release 19, is the explicit handling of NAUN3 device authentication. 3GPP in its specifications [43] notes that for a non-3GPP device connecting via a UE (the 5G-RG), the "non-3GPP device does not use NAS and is not authenticated by 5GC." While Release 19 provides methods to identify and apply QoS to such devices after they are somehow recognized by the 5G-RG, it does not standardize a secure authentication mechanism for these devices that is then directly linked to their service provisioning within the 5G system.

The implementation detailed in this chapter directly addresses this by:

1. Configuring FreeRADIUS as an EAP-TLS Authentication Server.
2. Setting up `hostapd` on the 5G-RG (`ue` VM) as an EAP Authenticator (relay).
3. Configuring `wpa_supplicant` on the NAUN3 device (`naun3` VM) for EAP-TLS.
4. Utilizing the custom interceptor service on the 5G-RG to listen for `CTRL-EVENT-EAP-SUCCESS` from `hostapd`.

This successful local EAP-TLS authentication, orchestrated by these standard tools and the custom `interceptor`, serves as the verified trigger for the 5G-RG to proceed with requesting a dedicated PDU Session (using the `clients` DNN) for the now-authenticated NAUN3 device. This provides a secure and verifiable onboarding step before 5G resources are allocated, a mechanism not explicitly detailed for such devices in the current 3GPP Release 19 flows for differentiated QoS.

In essence, while 3GPP Release 19 is building pathways for differentiated services for non-3GPP devices, the implemented framework demonstrates a practical and secure method for the initial authentication and individual session-based integration of these devices, which can then leverage the evolving QoS and policy mechanisms within the 5G system.

CHAPTER 5

Validation and Results Evaluation

This chapter presents the validation process and evaluates the performance and effectiveness of the implemented solution. Through a series of defined test scenarios and experiments, quantitative and qualitative results are gathered and analyzed. These findings are then critically assessed against the initial research objectives and compared with existing approaches discussed in the state of the art.

5.1 Methodology

To assess the feasibility, functionality, and effectiveness of the proposed solution, a series of tests were conducted within the simulated environment described in chapter 4. The overall validation approach was simulation-based testing, leveraging the orchestrated virtual machines and configured 5G components (Open5GS, UERANSIM) and local network services (`hostapd`, `dnsmasq`, and the custom `interceptor` application).

The validation focused on several key aspects of the system:

5.1.1 Overall Validation Approach

- **Simulation-Based Testing:** All validation activities were performed within the virtualized environment created using Vagrant, Open5GS, UERANSIM, FreeRADIUS, and the custom `interceptor` application. This allowed for controlled and replicable testing of the end-to-end solution.
- **Focus on Functional Correctness and Integration:** The primary goal was to verify that the proposed mechanisms for authentication, proxy identity creation (per-device PDU session), traffic mapping, and lifecycle management operate as designed.
- **Qualitative Security Assessment:** While not a formal security audit, the validation included observing whether the implemented security measures (EAP-TLS, traffic separation) were functioning as intended.

5.1.2 KPIs and Metrics for Evaluation

The evaluation of the framework centered on the following indicators and metrics, primarily assessed through functional testing and observation of system logs and behavior:

1. **Functional Correctness:**
   a) **NAUN3 Device Authentication Success:**
      - **Metric (Establishment) (ID-1):** Successful completion of the EAP-TLS authentication process for an NAUN3 device with the FreeRADIUS server, relayed by the 5G-RG (`hostapd` and `interceptor`).
      - **Metric (Timeliness) (ID-2):** General observation of the time taken for the end-to-end process: NAUN3 device EAP-TLS authentication and subsequent PDU session establishment and traffic segregation and mapping, until 5GC is reachable from the NAUN3.
      - **Verification:** Logs from `wpa_supplicant` (`naun3` VM), `hostapd` (5G-RG/`ue` VM), FreeRADIUS (`core` VM), and the custom `interceptor` application on the 5G-RG.
   b) **Dedicated PDU Session Establishment:**
      - **Metric (ID-3):** Successful establishment of a unique PDU session on the `clients` DNN by the 5G-RG for each successfully authenticated NAUN3 device.
      - **Verification:** Output of `nr-cli ps-list` command on the 5G-RG (`ue` VM); logs from Open5GS SMF and UPF on the `core` VM.
   c) **IP Address Allocation:**
      - **Metric (Local) (ID-4):** Successful assignment of a local IP address to the NAUN3 device by `dnsmasq` on the 5G-RG after EAP-TLS authentication.
      - **Metric (5GC) (ID-5):** Successful assignment of a 5GC IP address by the 5GC (SMF/UPF) to the dedicated PDU session for the NAUN3 device.
      - **Verification:** `ip addr` command output on the `naun3` VM; `dnsmasq` logs on the 5G-RG; `nr-cli ps-list` output on the 5G-RG; Open5GS SMF/UPF logs.
   d) **End-to-End Data Plane Connectivity:**
      - **Metric (ID-6):** Ability of an authenticated NAUN3 device to send and receive IP traffic to/from an external network via its dedicated PDU session.
      - **Verification:** Ping tests and simple data transfer from the `naun3` VM to a target beyond the UPF; packet captures (`tcpdump`) on NAUN3 LAN interface, 5G-RG's PDU session tunnel interface, and UPF interfaces.
   e) **Traffic Isolation and Mapping:**
      - **Metric (ID-7):** Confirmation that traffic from a specific NAUN3 device is routed exclusively through its dedicated PDU session and associated routing rules.

- **Verification:** Traffic captures on the 5G-RG using `tcpdump`; analysis of `iptables` counters, `ip rule` and `ip route` configurations on the 5G-RG during active traffic from one or more NAUN3 devices.

f) **Lifecycle Management Correctness:**

- **Metric (ID-8):** Successful de-authentication of an NAUN3 device and termination of its associated PDU session upon simulated disconnection/unreachability.
- **Verification:** Logs from the `interceptor` application, `hostapd`, and `dnsmasq`; `nr-cli ps-list` output showing PDU session release; verification of removal of `iptables` rules and `dnsmasq` permissions.

These metrics are also aimed to attest the following key aspects:

- **Security Aspects (Qualitative Observation):**
  - **EAP-TLS Authentication Integrity:** Observation of the complete EAP-TLS handshake and successful mutual authentication through detailed logs from involved components (`wpa_supplicant`, `hostapd`, FreeRADIUS).
  - **Traffic Segregation:** Confirmation via network monitoring and PDU session analysis that `backhaul` DNN traffic (e.g., RADIUS) remains logically separate from the `clients` DNN traffic (NAUN3 user plane data).
- **Resource Management (Observational):**
  - **PDU Session Correlation with authenticated NAUN3 devices:** The number of active PDU sessions on the `clients` DNN should directly correspond to the number of currently authenticated and connected NAUN3 devices, as tracked by the `interceptor` application.
- **System Stability and Robustness (Qualitative):**
  - **Handling Multiple Devices:** The ability of the `interceptor` application and the overall simulated system to manage sequential and concurrent connections and disconnections of multiple NAUN3 devices without instability.
  - **Error Handling:** Observation of error logging and any recovery mechanisms within the `interceptor` application in scenarios such as a failed PDU session establishment attempt or unexpected disconnection.

This validation methodology aims to provide a comprehensive assessment of the implemented solution's ability to meet its design goals, focusing on correct functionality and integration within the simulated 5G environment. The subsequent sections will detail the specific test scenarios designed and the evaluation of the results obtained.

5.2 Test Scenarios and Setup

To validate the different aspects of the proposed framework, a series of distinct test scenarios, or experiments, were designed and executed. These scenarios leveraged the fully configured simulation environment detailed in the chapter 4 and visible in Figure 5.1, which

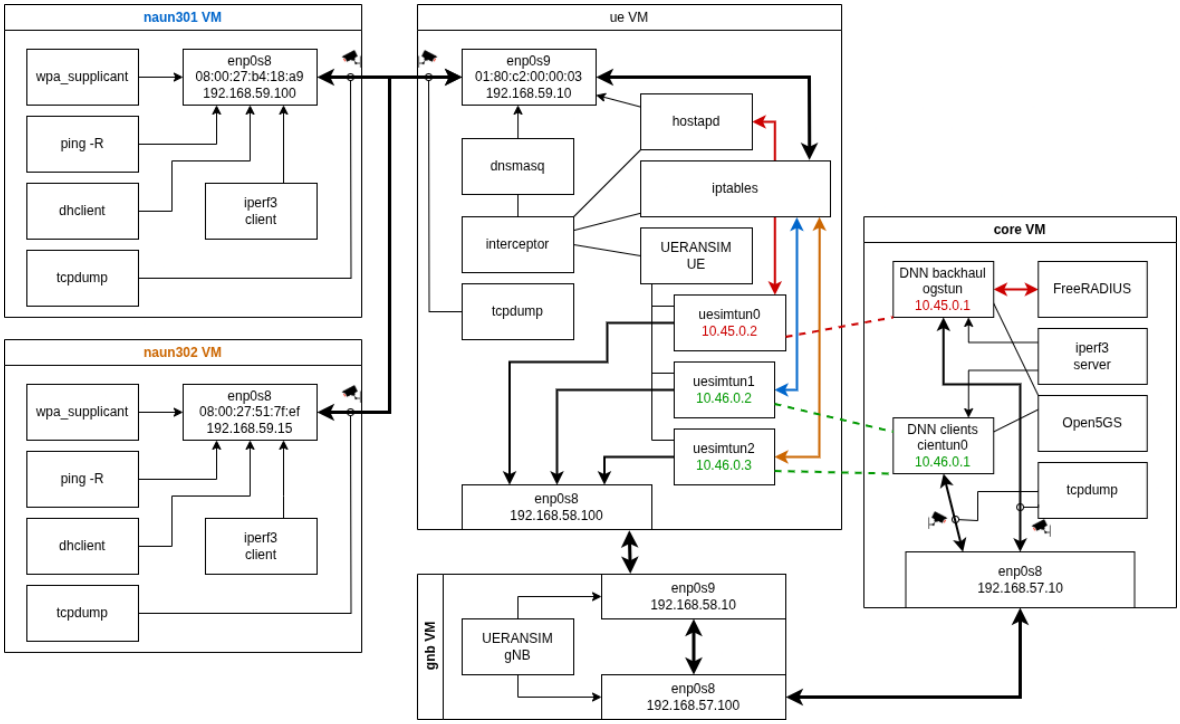


Figure 5.1: Fully emulated testing environment

includes the `core` VM (Open5GS, FreeRADIUS), `gnb` VM (UERANSIM gNB), `ue` VM (5G-RG with UERANSIM UE, `hostapd`, `dnsmasq`, and the custom `interceptor` application), and one or more `naun3` VMs (EAP supplicant).

Specific tools were employed for monitoring and verification in each scenario:

- **Log Analysis:** Reviewing logs from Open5GS NFs, UERANSIM components, FreeRADIUS, `hostapd`, `wpa_supplicant`, and the custom `interceptor` application was fundamental across all tests.
- **Packet Capture:** `tcpdump` and `tshark` (Wireshark Command Line Interface (CLI)) were used on various interfaces (NAUN3 LAN, 5G-RG's `backhaul` and `clients` PDU session interfaces `uesimtunX`, gNB interfaces) to inspect signaling and data plane traffic.
- **Network Utilities:** Standard Linux utilities like `ping` (with the `-R` record route option), `ip addr`, `ip route`, `ip rule`, `iptables -L -v -n -t mangle -t nat -t filter`, and UERANSIM's `nr-cli` were used for connectivity testing and state verification.
- **Traffic Generation and Throughput Measurement:** `iperf3` was used for generating controlled Transmission Control Protocol (TCP)/UDP network traffic between NAUN3 devices and a server on the `core` VM (simulating an N6-connected server) to test data plane throughput and routing.
- **Timestamping/Scripting:** Basic shell scripting was used to capture timestamps before and after key events (e.g., `wpa_supplicant` start and `dhclient` IP acquisition) to measure onboarding delay.

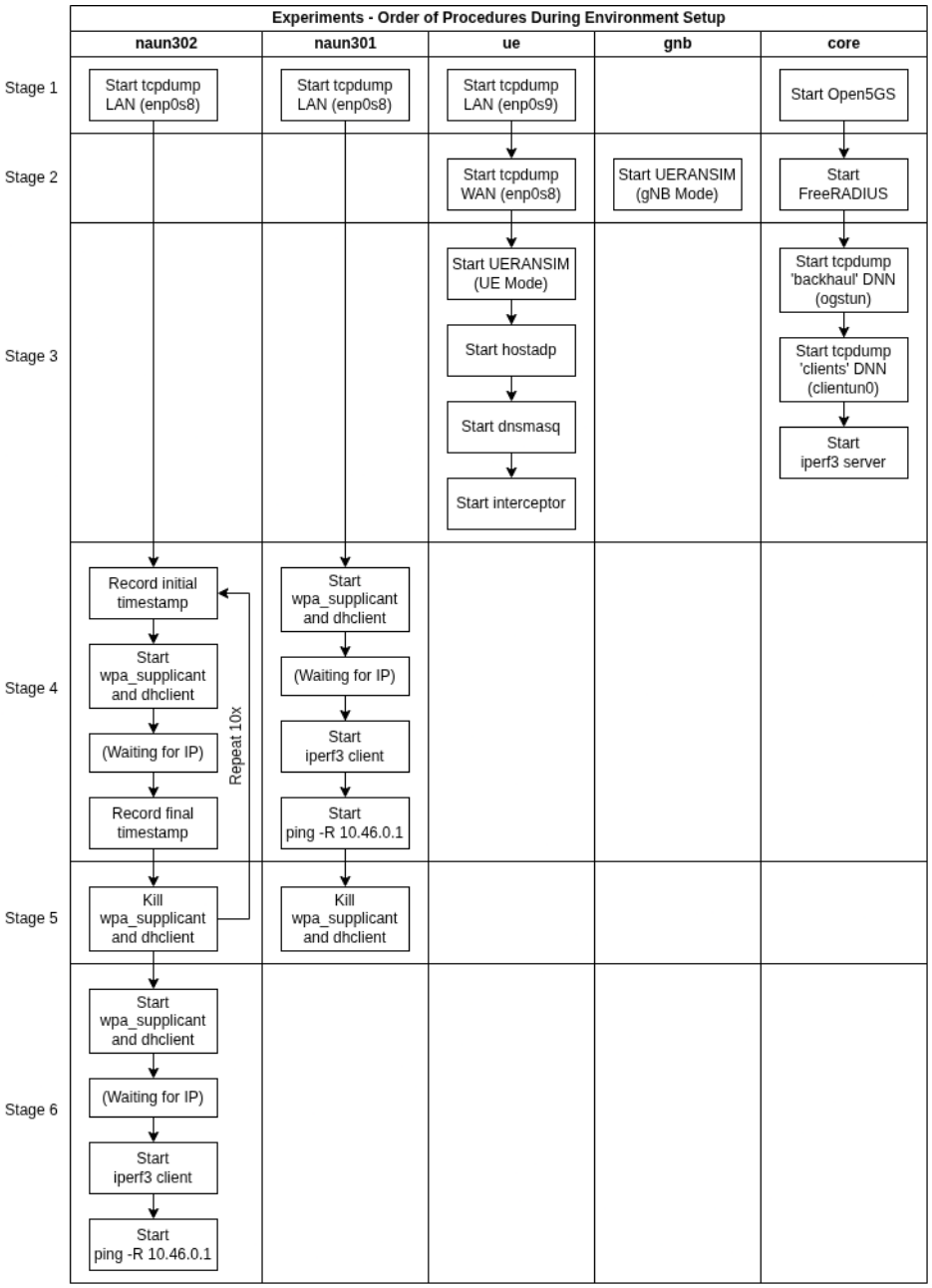


Figure 5.2: Order of Procedures During Environment Setup

The experiments will all utilize the same global procedure, as shown in Figure 5.2, but not all stage are relevant for each experiment. As such, the relevant steps of the procedure will be explicitly described in each experiment.

5.2.1 Experiment 1: Single NAUN3 Device Onboarding and Basic Connectivity

The objective was to verify the successful EAP-TLS authentication of a single NAUN3 device, the subsequent establishment of its dedicated PDU session on the `clients` DNN, local and 5GC IP address allocation, basic end-to-end data plane connectivity with path verification, and to measure the approximate onboarding delay.

Procedure:

1. Ensure all 5GC NFs, gNB, 5G-RG (including `hostapd` and `interceptor`), and FreeRADIUS are running. Start an `iperf3` server on the `core` VM listening on the IP address of its `clientun0` interface (e.g., `10.46.0.1`).
2. Capture traffic on relevant interfaces (`naun3` LAN, `ue` VM LAN, `uesimtunX` on `ue` VM, `clientun0` and `ogstun` on `core` VM) to observe data flow and NAT.
3. On the `naun3` VM, record a timestamp (`date +%s`). Immediately start the `wpa_supplicant` service.
4. Monitor the authentication process through logs on the `naun3` VM, `ue` VM (`hostapd`, `interceptor`), and `core` VM (FreeRADIUS).
5. Once `wpa_supplicant` indicates success and `dhclient` (run subsequently or as part of the script on `naun3`) obtains a local IP, record another timestamp. Calculate the difference to estimate onboarding delay.
6. Verify PDU session establishment for the `clients` DNN using `nr-cli ps-list` on the `ue` VM. Note the assigned 5GC IP address for this PDU session.
7. Verify local IP address assignment to the `naun3` VM via `ip addr` on the `naun3` VM.
8. Verify the application of `iptables` and `ip rule`/`ip route` rules on the `ue` VM specific to the authenticated NAUN3 device's MAC and PDU session ID.
9. From the `naun3` VM, initiate `ping -R <PDU_Session_Gateway_IP>` (e.g., `10.46.0.1`). Analyze the recorded route to confirm it passes through the NAUN3's local IP, then its assigned 5GC PDU session IP, and then to the target.
10. From the `naun3` VM, run an `iperf3` client connecting to the `iperf3` server on the `core` VM.

Metrics/Verification Points:

- Successful EAP-TLS authentication (logs) - Metric ID-1.
- Onboarding delay average of multiple successful onboardings(timestamp delta) - Metric ID-2.
- One new PDU session active on `clients` DNN for the 5G-RG, with a unique 5GC IP - Metric ID-3.
- NAUN3 device receives a local IP - Metric ID-4.
- Successful `ping` with recorded route showing NAT via the PDU session IP - Metric ID-5 and ID-7.
- Successful `iperf3` data transfer - Metric ID-6.
- Correct `iptables` and policy routing rules applied - Metric ID-7.

5.2.2 Experiment 2: Multi-Device Connectivity, Traffic Isolation, and PDU Session Mapping

In this experiment the goal was to verify that when multiple NAUN3 devices connect simultaneously, each gets a unique dedicated PDU session, their traffic is correctly mapped and isolated, and NAT occurs via their respective PDU session IPs.

Procedure:

1. Start multiple (two or more) `naun3` VMs (e.g., `naun301`, `naun302`), each configured with a client certificates for EAP-TLS.
2. Allow both devices to authenticate and establish their dedicated PDU sessions as per Experiment 1. Verify that two distinct `clients` PDU sessions are created by the 5G-RG, each with a unique 5GC IP address (e.g., `10.46.0.2` and `10.46.0.3`).
3. On the `ue` VM, verify that distinct sets of `iptables` and `ip rule`/`ip route` entries are created for each NAUN3 device, mapping each to its unique PDU session ID and tunnel interface.
4. Check `iptables` counters to confirm traffic from each NAUN3 device is routed through its distinct PDU session.
5. From `naun301`, execute `ping -R <PDU_Session_Gateway_IP>`. Note the recorded route, particularly the 5GC IP address assigned to `naun301`'s PDU session.
6. From `naun302`, execute `ping -R <PDU_Session_Gateway_IP>`. Note the recorded route, verifying it uses a *different* 5GC IP address assigned to `naun302`'s PDU session.
7. Start an `iperf3` server on the `core` VM.
8. Simultaneously (or sequentially) run `iperf3` clients from `naun301` and `naun302` to the server on the `core` VM.
9. Monitor traffic on the `ue` VM's `uesimtunX` interfaces and check `iptables` counters to confirm traffic from each NAUN3 device is routed through its distinct PDU session.
10. On the `core` VM (`iperf3` server logs), verify that connections are received from the distinct 5GC IP addresses assigned to each NAUN3's PDU session.

Metrics/Verification Points:

- Each NAUN3 device establishes its own unique PDU session on the `clients` DNN with a distinct 5GC IP - Metric ID-3 and ID-7.
- `ping -R` from each NAUN3 shows a path *NATted* through its unique PDU session IP - Metric ID-5 and ID-7.
- `iperf3` server logs show connections from distinct PDU session IPs - Metric ID-7.
- `iptables` and routing rules correctly isolate traffic per device - Metric ID-7.

5.2.3 Experiment 3: Lifecycle Management (Device Disconnection and Resource Cleanup)

In order to verify that when an NAUN3 device disconnects or becomes unreachable, its local authentication is revoked, its dedicated PDU session is terminated, and associated network resources (IP addresses, routing rules) are correctly cleaned up by the `interceptor` the following procedure was followed.

1. Successfully onboard a single NAUN3 device as per Experiment 1. Verify its PDU session is active and traffic flows.
2. Simulate device disconnection:
   - **Option A (Graceful):** Stop the `wpa_supplicant` service on the `naun3` VM.
   - **Option B (Abrupt):** Power off or disconnect the network interface of the `naun3` VM.
3. Monitor the `interceptor` logs on the `ue` VM for detection of device unreachability and initiation of cleanup procedures.
4. Verify the following cleanup actions occur:
   - `hostapd` deauthenticates the client (logs).
   - The `interceptor` removes the device's MAC from `/etc/allowed-macs.conf` and restarts `dnsmasq`. In normal circumstances, DHCP lease renovation does not constitute a relevant event to the system and as such does not trigger any response from the `interceptor`, mainly because traffic routing is done using the NAUN3 device MAC address. In addition, when first launching the `interceptor` application, the user can define the duration of the DHCP leases.
   - The `interceptor` removes the specific `iptables` and `ip rule`/`ip route` entries for the device.
   - The `interceptor` initiates a PDU session release for the `clients` DNN using `nr-cli`.
   - Verify using `nr-cli ps-list` that the PDU session is terminated.
   - Verify in Open5GS SMF/UPF logs that the session and associated resources are released.

Metrics/Verification Points:

- Detection of device disconnection by the `interceptor` - Metric ID-8.
- Successful local deauthentication - Metric ID-8.
- Removal of DHCP permission - Metric ID-8.
- Correct removal of all associated `iptables` and routing rules - Metric ID-7 and ID-8.
- Successful PDU session termination in UERANSIM and Open5GS - Metric ID-3 and ID-8.
- Internal state of the `interceptor` (e.g., `allowedDevices` map) reflects the device removal.

5.2.4 Experiment 4: Security Aspects Observation (Qualitative)

To qualitatively observe key security aspects of the implemented solution, such as the EAP-TLS handshake and traffic segregation, we performed the following:

1. During Experiment 1 (NAUN3 Device Onboarding):
   - Use `tshark` or `tcpdump` on the `ue` VM's LAN interface to capture the EAPOL and EAP-TLS handshake.

- Use `tshark` or `tcpdump` on the `ue` VM's `backhaul` PDU session interface and on the `core` VM's interface connected to the `ue` VM's `backhaul` to capture RADIUS traffic.

2. During Experiment 2 (Multiple Devices):
   - Observe the source and destination IPs of the RADIUS traffic on the `backhaul` PDU session.
   - Observe the source and destination IPs of the NAUN3 user plane traffic on the respective `clients` PDU sessions.

Metrics/Verification Points:

- Observation of a complete and successful EAP-TLS handshake - Metric ID-1.
- Confirmation that RADIUS traffic is transported over the `backhaul` PDU session - Metric ID-1 .
- Confirmation that user plane traffic for each NAUN3 device is transported over its distinct `clients` PDU session - Metric ID-3 and ID-7.

These experiments are designed to provide a holistic view of the system's functionality, its ability to manage multiple devices correctly, handle their lifecycle, and maintain basic security and traffic segregation principles, incorporating the specific types of data you've collected.

5.3 Functional Validation Results

This section presents the results obtained from executing the test scenarios described previously, demonstrating the functional correctness of the proposed authentication and identity management mechanisms. The evidence is drawn from system logs, network interface states, routing table configurations, and packet captures.

5.3.1 NAUN3 Device Authentication and Onboarding:

The primary test involved onboarding NAUN3 devices (`naun301`, `naun302`) one after another.

- **EAP-TLS Authentication:** Logs from `wpa_supplicant` on each NAUN3 device (see Listing 5.1) confirmed the initiation and successful completion of the EAP-TLS handshake (`"CTRL-EVENT-EAP-SUCCESS EAP authentication completed successfully"`). Correspondingly, the `hostapd` log on the 5G-RG (`ue` VM) showed the EAP exchange, including the relay of messages to the RADIUS server and the reception of Access-Accept (see Listing 5.2). The `interceptor` (see Listing 5.3) captured the `CTRL-EVENT-EAP-SUCCESS` from `hostapd`, indicating its awareness of the successful local authentication.

```
(...)
584| 1748702590.706948: enp0s8: CTRL-EVENT-EAP-SUCCESS EAP authenticati
on completed successfully
585| 1748702590.706998: EAPOL: IEEE 802.1X for plaintext connection; no
EAPOL-Key frames required
586| 1748702590.707047: enp0s8: WPA: EAPOL processing complete
```

```
587| 1748702590.707096: enp0s8: Cancelling authentication timeout
586| 1748702590.707147: enp0s8: State: ASSOCIATED -> COMPLETED
586| 1748702590.707201: enp0s8: CTRL-EVENT-CONNECTED - Connection to 01:
80:c2:00:00:03 completed [id=0 id_str=]
(...)
```

Listing 5.1: wpa_supplicant successful authentication

```
(...)
802| 1748702590.717640: EAP: EAP entering state SUCCESS2
803| 1748702590.717722: enp0s9: CTRL-EVENT-EAP-SUCCESS2 08:00:27:b4:18:a
9
804| 1748702590.717775: CTRL_IFACE monitor send /tmp/interceptor_7025.so
ck\x00
805| 1748702590.717923: IEEE 802.1X: 08:00:27:b4:18:a9 BE_AUTH entering
state SUCCESS
806| 1748702590.718200: 1748702590.718200: enp0s9: STA 08:00:27:b4:18:a9
IEEE 802.1X: Sending EAP Packet (identifier 224)
807| 1748702590.718452: IEEE 802.1X: 08:00:27:b4:18:a9 AUTH_PAE entering
state AUTHENTICATED
808| 1748702590.718565: enp0s9: AP-STA-CONNECTED 08:00:27:b4:18:a9
(...)
```

Listing 5.2: `hostapd` successful authentication

- **Dedicated PDU Session Establishment:** Following each successful EAP-TLS authentication (see Listing 5.4), the `interceptor` log shows the initiation of a new PDU session request for the `clients` DNN (see Listing 5.4), and finally allows the device to request IP addresses via DHCP (see Listing 5.5). The UERANSIM tool `nr-cli ps-list` on the 5G-RG, corroborates this (see Listing 5.3):
  - Initially (14:43:09 Universal Time Code (UTC)), only `PDU Session1` (DNN: `backhaul`, IP: `10.45.0.2`) is active.
  - After `naun301` (MAC `08:00:27:b4:18:a9`) authenticates (around 14:43:10 UTC in `interceptor` log), `PDU Session2` (DNN: `clients`, IP: `10.46.0.2`) becomes active by 14:43:30 UTC.
  - After `naun302` (MAC `08:00:27:51:7f:ef`) authenticates (around 14:44:08 UTC in `interceptor` log), `PDU Session3` (DNN: `clients`, IP: `10.46.0.3`) becomes active by 14:44:30 UTC.

```
(...)
19| [DEBUG] 2025/05/31 14:43:10  Auth success for 08:00:27:b4:18:a9
(...)
```

Listing 5.3: `interceptor` captures authentication success

```
(...)
19| [DEBUG] 2025/05/31 14:43:10  Auth success for 08:00:27:b4:18:a9
20| [DEBUG] 2025/05/31 14:43:10  Requesting PDU for 08:00:27:b4:18:a9 (I
```

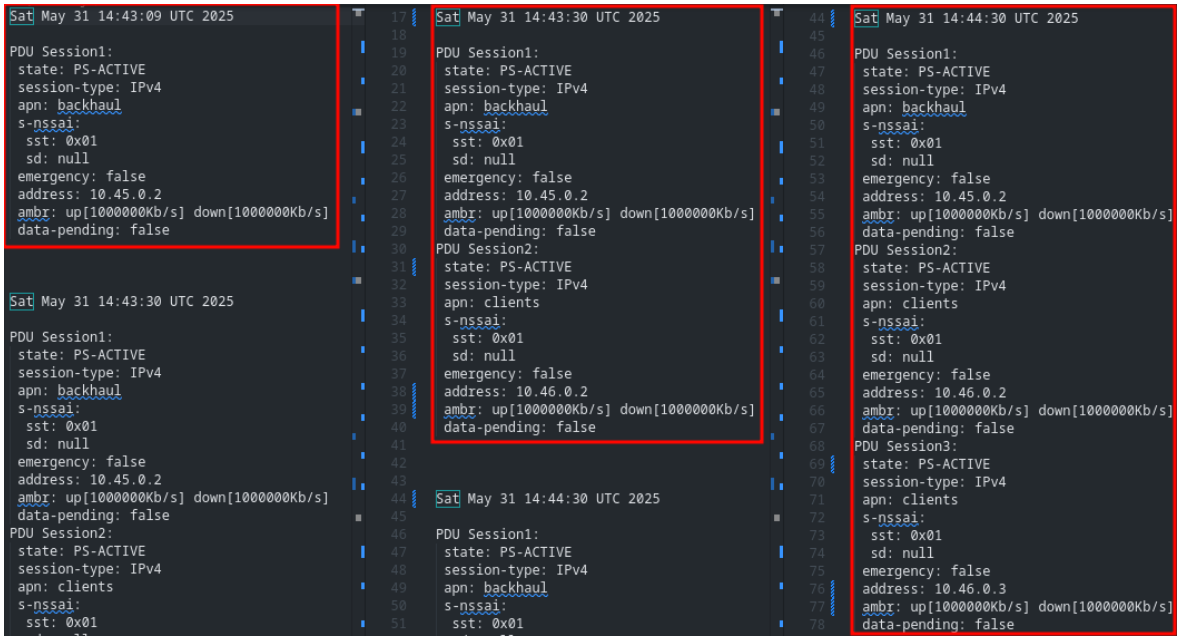


Figure 5.3: PDU Session via `nr-cli ps-list`

```
MSI: imsi-999700000000001)
21| [DEBUG] 2025/05/31 14:43:10  IMSI imsi-999700000000001 establishing
...
22| [DEBUG] 2025/05/31 14:43:10  IMSI imsi-999700000000001 requested. O
utput: PDU session establishment procedure triggered. Waiting activatio
n...
(...)
28| [DEBUG] 2025/05/31 14:43:28  PDU ID 2 IMSI imsi-999700000000001 ACT
IVE (State: PS-ACTIVE, Addr: 10.46.0.2).
29| [DEBUG] 2025/05/31 14:43:28  PDU Session ID 2, constructed PDU Inte
rface: uesimtun1
(...)
```

Listing 5.4: `interceptor` requests PDU session

```
(...)
48| [DEBUG] 2025/05/31 14:43:28  MAC 08:00:27:b4:18:a9 added to /etc/dns
masq.d/allowed-macs.conf.
(...)
```

Listing 5.5: `interceptor` MAC address permited for IP attribution

- **Network Interface Creation:** The `ue` (see Listing 5.6) log confirms the dynamic creation of network interfaces on the 5G-RG for each `clients` PDU session. `uesimtun0` (IP `10.45.0.2`) corresponds to the `backhaul` session. `uesimtun1` (IP `10.46.0.2`) appears after `naun301` connects, and `uesimtun2` (IP `10.46.0.3`) appears after `naun302` connects.

```
Sat May 31 14:40:08 UTC 2025
5: uesimtun0: <POINTOPOINT,PROMISC,NOTRAILERS,UP,LOWER_UP> mtu 1400 ...
    inet 10.45.0.2/24 scope global uesimtun0
```

```
Sat May 31 14:43:30 UTC 2025
5: uesimtun0: <POINTOPOINT,PROMISC,NOTRAILERS,UP,LOWER_UP> mtu 1400 ...
    inet 10.45.0.2/24 scope global uesimtun0
6: uesimtun1: <POINTOPOINT,PROMISC,NOTRAILERS,UP,LOWER_UP> mtu 1400 ...
    inet 10.46.0.2/24 scope global uesimtun1

Sat May 31 14:44:30 UTC 2025
5: uesimtun0: <POINTOPOINT,PROMISC,NOTRAILERS,UP,LOWER_UP> mtu 1400 ...
    inet 10.45.0.2/24 scope global uesimtun0
6: uesimtun1: <POINTOPOINT,PROMISC,NOTRAILERS,UP,LOWER_UP> mtu 1400 ...
    inet 10.46.0.2/24 scope global uesimtun1
7: uesimtun2: <POINTOPOINT,PROMISC,NOTRAILERS,UP,LOWER_UP> mtu 1400 ...
    inet 10.46.0.3/24 scope global uesimtun2
(...)
```

Listing 5.6: Network interfaces created by UERANSIM to bind to PDU Sessions

- **IP Address Allocation:** The NAUN3 devices successfully obtained local IP addresses from `dnsmasq` on the 5G-RG after authentication (see Listing 5.7 and 5.8). The 5GC assigned unique IPs (`10.46.0.2`, `10.46.0.3`) to their respective PDU sessions, as confirmed by Figures 5.3 and 5.6.

```
PING 10.46.0.1 (10.46.0.1) 56(124) bytes of data.
64 bytes from 10.46.0.1: icmp_seq=1 ttl=63 time=1.52 ms
RR:   192.168.59.100
    10.46.0.2
    10.46.0.1
    10.46.0.1
    192.168.59.10
    192.168.59.100
(...)
```

Listing 5.7: `naun301 ping -R` displaying route to the 5GC

```
PING 10.46.0.1 (10.46.0.1) 56(124) bytes of data.
64 bytes from 10.46.0.1: icmp_seq=1 ttl=63 time=2.31 ms
RR:   192.168.59.15
    10.46.0.3
    10.46.0.1
    10.46.0.1
    192.168.59.10
    192.168.59.15
(...)
```

Listing 5.8: `naun3012 ping -R` displaying route to the 5GC

```
Measured between first EAP Identity Request to last DHCP ACK
Attempt #1 0.005833 to 37.134592
Attempt #2 167.449142 to 196.922320
Attempt #3 327.151600 to 367.484920
Attempt #4 497.703455 to 530.219903
```

```
Attempt #5 660.442262 to 689.500301
Attempt #6 819.773031 to 858.609561
Attempt #7 988.818362 to 1025.744125
Attempt #8 1155.971666 to 1187.690164
Attempt #9 1317.914145 to 1349.066786
Attempt #10 1479.301829 to 1503.792846
```

Listing 5.9: Timestamps from traffic captures at the `naun301` during authentication and IP atribution

- **Onboarding Delay:** The onboarding delay was measured by repeating the onboarding process ten consecutive times using the `naun301` device. For each of the ten attempts, timestamps from traffic captures were used to determine the interval from the first EAP *Identity Request* to the final DHCP ACK (raw data in Listing 5.9).This procedure yielded an average onboarding delay of 33.1634193 seconds with a standard deviation of 5.0077937 seconds for this measured interval. This duration signifies the time taken from the initial EAP exchange (which is part of the authentication sequence initiated by `wpa_supplicant`) until the NAUN3 device is fully authenticated, receives its local IP address (confirmed by the DHCP ACK), with its dedicated PDU Session. This comprehensive measured period includes the EAP-TLS handshake, all RADIUS communications, the PDU session establishment orchestrated by the `interceptor` via `nr-cli`, and the final DHCP lease acquisition on the local network.

5.3.2 End-to-End Connectivity and Path Verification

Ping tests with the record route option (`-R`) were conducted from `naun301` and `naun302` (see Listings 5.7 and 5.8) to the clients DNN gateway IP on the `core` VM (`10.46.0.1`).

- The recorded route when (naun301) pings the 5GC gateway is, `192.168.59.100` (`naun301` local IP) -> `10.46.0.2` (PDU Session 2 IP for `naun302`) -> `10.46.0.1` (Target). This clearly demonstrates that traffic from `naun301` is *NATted* using the IP address of its dedicated PDU session.
- The recorded route is `naun302` pings the 5GC gateway is, `192.168.59.15` (`naun302` local IP) -> `10.46.0.3` (PDU Session 3 IP for `naun302`) -> `10.46.0.1` (Target). This confirms that `naun302`'s traffic is also *NATted*, but crucially, via its own distinct PDU session IP.

5.3.3 Traffic Isolation and Correct PDU Session Mapping (Multiple Devices)

```
Accepted connection from 10.46.0.2, port 36080
[  6] local 10.46.0.1 port 5201 connected to 10.46.0.2 port 36094
[  9] local 10.46.0.1 port 5201 connected to 10.46.0.2 port 36096
(...)
Accepted connection from 10.46.0.3, port 46222
[  7] local 10.46.0.1 port 5201 connected to 10.46.0.3 port 46232
[ 10] local 10.46.0.1 port 5201 connected to 10.46.0.3 port 46236
```

Listing 5.10: `iperf3` server session receiveing from both clients at `naun301` and `naun302` via it's seperate PDU channels

The `iperf3` tests further validated traffic isolation and mapping. As seen in the Listing 5.10, it shows the `iperf3` server on `10.46.0.1` accepting connections from two different source IPs: `10.46.0.2` and `10.46.0.3`. These correspond to the unique PDU session IPs assigned to `naun301` and `naun302` respectively. This confirms that traffic from each NAUN3 device is correctly mapped to its dedicated PDU session and is identifiable by this unique PDU session IP.

```
Sat May 31 14:43:30 UTC 2025
Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
    state RELATED,ESTABLISHED
    0     0 ACCEPT     all  --  enp0s9 uesimtun1  0.0.0.0/0            0.0.0.0/0
    MAC08:00:27:b4:18:a9 mark match 0x2

Sat May 31 14:44:30 UTC 2025
Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
43701   58M ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
state RELATED,ESTABLISHED
    6   456 ACCEPT     all  --  enp0s9 uesimtun1  0.0.0.0/0            0.0.0.0/0
    MAC08:00:27:b4:18:a9 mark match 0x2
    0     0 ACCEPT     all  --  enp0s9 uesimtun2  0.0.0.0/0            0.0.0.0/0
    MAC08:00:27:51:7f:ef mark match 0x3
```

Listing 5.11: `iptables` mapping rules and tables for segregating traffic

Also, Listing 5.11 shows the dynamic application of `iptables FORWARD` rules and `ip rule/ip route` entries. For instance, at 14:43:30 UTC (after `naun301` connects), rules are present for MAC `08:00:27:b4:18:a9` (`naun301`) to use PDU ID 2 (interface `uesimtun1`). By 14:44:30 UTC (after `naun302` connects), additional rules appear for MAC `08:00:27:51:7f:ef` (`naun302`) to use PDU ID 3 (interface `uesimtun2`), while rules for PDU ID 2 remain. This demonstrates the per-device rule application.

The Wireshark capture in Figure 5.4 visually confirms this: traffic on the 5G-RG's LAN interface shows the NAUN3's local IP (e.g., `192.168.59.100`), while on the corresponding PDU session tunnel interface (`uesimtunX`), the source IP is the PDU session's 5GC-assigned IP (e.g., `10.46.0.3`)

UE's LAN interface during NAUN3 iperf session with Core

No.	Time	Source	Destination	Protocol	Length	Info
20	69.091542	192.168.59.100	10.46.0.1	TCP	70	33950 → 5201 [PSH, ACK] Seq=38 Ack=2 Win=64256 Len=4 TSval=4155962625 TSecr=30
21	69.138713	10.46.0.1	192.168.59.100	TCP	66	5201 → 33950 [ACK] Seq=2 Ack=42 Win=65152 Len=0 TSval=3022644133 TSecr=4155962
22	69.139700	192.168.59.100	10.46.0.1	TCP	187	33950 → 5201 [PSH, ACK] Seq=42 Ack=2 Win=64256 Len=121 TSval=4155962673 TSecr=
23	69.143595	10.46.0.1	192.168.59.100	TCP	66	5201 → 33950 [ACK] Seq=2 Ack=163 Win=65152 Len=0 TSval=3022644138 TSecr=415596
24	69.143976	10.46.0.1	192.168.59.100	TCP	67	5201 → 33950 [PSH, ACK] Seq=2 Ack=163 Win=65152 Len=1 TSval=3022644138 TSecr=4
25	69.145292	192.168.59.100	10.46.0.1	TCP	74	33960 → 5201 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=4155962678 T
26	69.149036	10.46.0.1	192.168.59.100	TCP	74	5201 → 33960 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3
27	69.150211	192.168.59.100	10.46.0.1	TCP	66	33960 → 5201 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=4155962683 TSecr=30226441
28	69.150212	192.168.59.100	10.46.0.1	TCP	103	33960 → 5201 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=37 TSval=4155962683 TSecr=30
29	69.153132	192.168.59.100	10.46.0.1	TCP	74	33974 → 5201 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=4155962686 T
30	69.157927	10.46.0.1	192.168.59.100	TCP	66	5201 → 33960 [ACK] Seq=1 Ack=38 Win=65152 Len=0 TSval=3022644149 TSecr=4155962
31	69.158864	10.46.0.1	192.168.59.100	TCP	74	5201 → 33974 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3
32	69.160177	192.168.59.100	10.46.0.1	TCP	66	33974 → 5201 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=4155962693 TSecr=30226441
33	69.162107	192.168.59.100	10.46.0.1	TCP	103	33974 → 5201 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=37 TSval=4155962694 TSecr=30
34	69.171341	10.46.0.1	192.168.59.100	TCP	66	5201 → 33974 [ACK] Seq=1 Ack=38 Win=65152 Len=0 TSval=3022644161 TSecr=4155962
35	69.171975	10.46.0.1	192.168.59.100	TCP	1514	5201 → 33974 [ACK] Seq=1 Ack=38 Win=65152 Len=1448 TSval=3022644163 TSecr=4155
36	69.172322	10.46.0.1	192.168.59.100	TCP	1514	5201 → 33974 [PSH, ACK] Seq=1449 Ack=38 Win=65152 Len=1448 TSval=3022644163 TS
37	69.172616	192.168.59.100	10.46.0.1	TCP	66	33974 → 5201 [ACK] Seq=38 Ack=1449 Win=64128 Len=0 TSval=4155962706 TSecr=3022
38	69.173064	192.168.59.100	10.46.0.1	TCP	66	33974 → 5201 [ACK] Seq=38 Ack=2897 Win=63488 Len=0 TSval=4155962706 TSecr=3022
39	69.173339	10.46.0.1	192.168.59.100	TCP	1514	5201 → 33974 [ACK] Seq=2897 Ack=38 Win=65152 Len=1448 TSval=3022644163 TSecr=4

UE's PDU interface during NAUN3 iperf session with Core

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.46.0.3	10.46.0.1	TCP	60	38846 → 5201 [SYN] Seq=0 Win=65280 Len=0 MSS=1360 SACK_PERM TSval=4156408742 T
2	0.004966	10.46.0.1	10.46.0.3	TCP	60	5201 → 38846 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3
3	0.006434	10.46.0.3	10.46.0.1	TCP	52	38846 → 5201 [ACK] Seq=1 Ack=1 Win=65280 Len=0 TSval=4156408749 TSecr=30230902
4	0.006783	10.46.0.3	10.46.0.1	TCP	89	38846 → 5201 [PSH, ACK] Seq=1 Ack=1 Win=65280 Len=37 TSval=4156408749 TSecr=30
5	0.011515	10.46.0.1	10.46.0.3	TCP	52	5201 → 38846 [ACK] Seq=1 Ack=38 Win=65152 Len=0 TSval=3023090220 TSecr=4156408
6	0.011782	10.46.0.1	10.46.0.3	TCP	53	5201 → 38846 [PSH, ACK] Seq=1 Ack=38 Win=65152 Len=1 TSval=3023090220 TSecr=41
7	0.012958	10.46.0.3	10.46.0.1	TCP	52	38846 → 5201 [ACK] Seq=38 Ack=2 Win=65280 Len=0 TSval=4156408756 TSecr=3023090
8	0.013248	10.46.0.3	10.46.0.1	TCP	56	38846 → 5201 [PSH, ACK] Seq=38 Ack=2 Win=65280 Len=4 TSval=4156408756 TSecr=30
9	0.064158	10.46.0.1	10.46.0.3	TCP	52	5201 → 38846 [ACK] Seq=2 Ack=42 Win=65152 Len=0 TSval=3023090272 TSecr=4156408
10	0.065427	10.46.0.3	10.46.0.1	TCP	173	38846 → 5201 [PSH, ACK] Seq=42 Ack=2 Win=65280 Len=121 TSval=4156408808 TSecr=
11	0.071796	10.46.0.1	10.46.0.3	TCP	52	5201 → 38846 [ACK] Seq=2 Ack=163 Win=65152 Len=0 TSval=3023090278 TSecr=415640
12	0.072540	10.46.0.1	10.46.0.3	TCP	53	5201 → 38846 [PSH, ACK] Seq=2 Ack=163 Win=65152 Len=1 TSval=3023090278 TSecr=4
13	0.073871	10.46.0.3	10.46.0.1	TCP	60	38848 → 5201 [SYN] Seq=0 Win=65280 Len=0 MSS=1360 SACK_PERM TSval=4156408816 T
14	0.077148	10.46.0.1	10.46.0.3	TCP	60	5201 → 38848 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3
15	0.078473	10.46.0.3	10.46.0.1	TCP	52	38848 → 5201 [ACK] Seq=1 Ack=1 Win=65280 Len=0 TSval=4156408821 TSecr=30230902
16	0.078510	10.46.0.3	10.46.0.1	TCP	89	38848 → 5201 [PSH, ACK] Seq=1 Ack=1 Win=65280 Len=37 TSval=4156408821 TSecr=30
17	0.081555	10.46.0.3	10.46.0.1	TCP	60	38864 → 5201 [SYN] Seq=0 Win=65280 Len=0 MSS=1360 SACK_PERM TSval=4156408824 T
18	0.085758	10.46.0.1	10.46.0.3	TCP	52	5201 → 38848 [ACK] Seq=1 Ack=38 Win=65152 Len=0 TSval=3023090291 TSecr=4156408
19	0.086344	10.46.0.1	10.46.0.3	TCP	60	5201 → 38864 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3
20	0.087610	10.46.0.3	10.46.0.1	TCP	52	38864 → 5201 [ACK] Seq=1 Ack=1 Win=65280 Len=0 TSval=4156408830 TSecr=30230902

Figure 5.4: NAUN3 to 5GC `iperf3` session, captured at the 5G-RG showing the mapping between the local address and PDU session address

5.3.4 Lifecycle Management (Device Disconnection)

The logs demonstrate correct resource cleanup when `naun301` (associated with PDU Session2, IP `10.46.0.2`, interface `uesimtun1`) disconnects:

```
(...)
91 | [DEBUG] 2025/05/31 14:44:49  Tracked device 10.46.0.2 (MAC: 08:00:27:b4:
18:a9,State: REACHABLE) no longer in ARP list. Scheduling for forget.
(...)
95 | [DEBUG] 2025/05/31 14:44:49  Removing MAC 08:00:27:b4:18:a9 line: 'dhcp-
host=08:00:27:b4:18:a9,2m,set:known'
(...)
99 | [DEBUG] 2025/05/31 14:44:49  Removing 6 stored rules for MAC 08:00:27:b4
:18:a9
(...)
110| [DEBUG] 2025/05/31 14:44:49  Rule removal process completed for MAC 08:
00:27:b4:18:a9. Successfully removed/verified absent: 6 of 6.
(...)
113| [DEBUG] 2025/05/31 14:44:49  ReleasePDUSession: PDU ID 2 IMSI imsi-99970
0000000001 released. Output: PDU session release procedure(s) triggered
(...)
115| [DEBUG] 2025/05/31 14:44:49  HostapdInterceptor: Sending DEAUTH for 08:
00:27:b4:18:a9
(...)
```

Listing 5.12: `naun301` disconnected and 5G-RG's `interceptor` proceeds to deauthenticating it removing traffic mapping rules for it and releasing it's dedicated PDU session

- `interceptor` (around 14:44:49 UTC) logs, according to Listing 5.12: `"Tracked device 10.46.0.2 (MAC: 08:00:27:b4:18:a9, State: REACHABLE) no longer in ARP list. Scheduling for forget."` This is followed by logs indicating removal from `allowedDevices`, deauthentication via `hostapd`, removal of `iptables` rules, and PDU session release for PDU ID 2.

```
Sat May 31 14:44:51 UTC 2025
PDU Session1:
 state: PS-ACTIVE
 session-type: IPv4
 apn: backhaul
 s-nssai:
  sst: 0x01
  sd: null
 emergency: false
 address: 10.45.0.2
 ambr: up[1000000Kb/s] down[1000000Kb/s]
 data-pending: false
PDU Session3:
 state: PS-ACTIVE
 session-type: IPv4
 apn: clients
 s-nssai:
  sst: 0x01
  sd: null
 emergency: false
 address: 10.46.0.3
 ambr: up[1000000Kb/s] down[1000000Kb/s]
 data-pending: false
```

Listing 5.13: PDU session listing from UERANSIM

- According to Listing 5.13, at 14:44:51 UTC, PDU Session2 (IP `10.46.0.2`) is no longer listed, while PDU Session1 (`backhaul`) and PDU Session3 (for `naun302`, IP `10.46.0.3`) remain active.

```
Sat May 31 14:44:51 UTC 2025
5: uesimtun0: <POINTOPOINT,PROMISC,NOTRAILERS,UP,LOWER_UP> mtu 1400 ...
    inet 10.45.0.2/24 scope global uesimtun0
7: uesimtun2: <POINTOPOINT,PROMISC,NOTRAILERS,UP,LOWER_UP> mtu 1400 ...
    inet 10.46.0.3/24 scope global uesimtun2
(...)
```

Listing 5.14: PDU binded network interfaces after `uesimtun1` removal

- Listing 5.14 also confirms that by 14:44:51 UTC, the `uesimtun1` interface (associated with `10.46.0.2`) is gone, while `uesimtun0` and `uesimtun2` persist.

```
Sat May 31 14:44:51 UTC 2025
Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source       destination
97585  124M ACCEPT     all  --  *      *       0.0.0.0/0    0.0.0.0/0
```

```
state RELATED,ESTABLISHED
    4   304 ACCEPT     all  --  enp0s9 uesimtun2  0.0.0.0/0    0.0.0.0/0
    MAC08:00:27:51:7f:ef mark match 0x3
```

Listing 5.15: Mapping rules after `naun301` disconnect and PDU Session2 release

- Lastly, Listing 5.15 also hows that at 14:44:51 UTC, the `iptables FORWARD` rule for MAC `08:00:27:b4:18:a9` (PDU ID 2) has been removed, while the rule for MAC `08:00:27:51:7f:ef` (PDU ID 3) remains.

These results collectively demonstrate that the proposed mechanisms for NAUN3 device authentication, proxy identity creation via dedicated PDU sessions, per-device traffic mapping and NAT, and lifecycle management function correctly within the simulated environment.

5.4 Security Evaluation

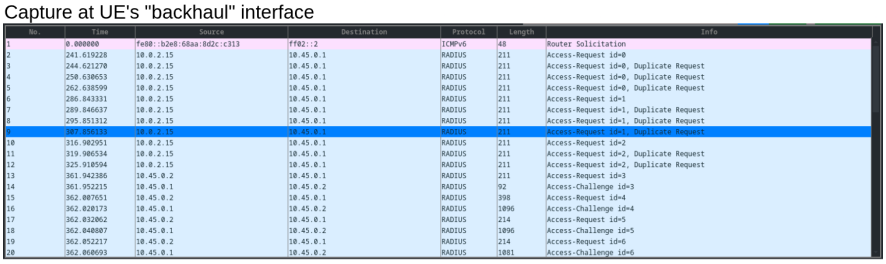
The security aspects of the solution were evaluated qualitatively based on the implemented mechanisms and observations from the test scenarios.

- **EAP-TLS Authentication Integrity:** The logs from `wpa_supplicant` in the NAUN3, `hostapd`, and FreeRADIUS consistently show the successful completion of the EAP-TLS handshake. This includes the exchange of certificates and the mutual verification steps inherent to the protocol. For instance, FreeRADIUS logs details like `"eap_tls: (TLS) Connection Established"` and `"eap: Sending EAP Success"`. This provides confidence that the local authentication of NAUN3 devices is cryptographically secured as per EAP-TLS standards.
- **Traffic Segregation (Control vs. User Plane):**
  - In Figure 5.5 a Wireshark capture clearly shows RADIUS packets (UDP port 1812), which carry the EAP authentication messages, being exchanged between the 5G-RG's `backhaul` PDU session IP (`10.45.0.2`) and the FreeRADIUS server's IP (`10.45.0.1`). This confirms that sensitive authentication control plane traffic is isolated to the dedicated `backhaul` DNN.
  - The `hostapd` log (see Listing 5.16) further details the RADIUS exchange, initially attempting to send from a default system IP (e.g., `10.0.2.15`) and failing (evident by retransmissions), then succeeding once the `own_ip_addr` (`10.45.0.2` - the `backhaul` PDU session IP) is correctly used. This highlights the correct binding and use of the `backhaul` PDU for RADIUS.

```
(...)
25 | 1748702388.755637: RADIUS local address: 10.0.2.15:42579
(...)
216| 1748702566.372520: RADIUS local address: 10.45.0.2:50371
(...)
```

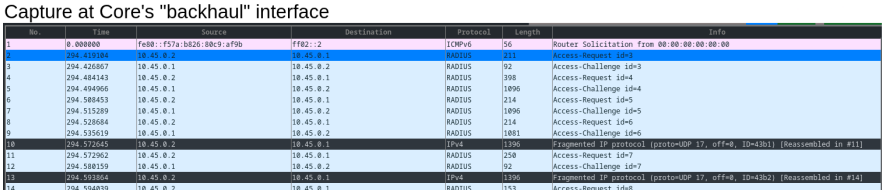
Listing 5.16: `hostapd` failing due to wrong local address

Capture at UE's "backhaul" interface



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::b2e8:68aa:8d2c:c313	ff02::2	ICMPv6	48	Router Solicitation
2	241.619228	10.0.2.15	10.45.0.1	RADIUS	211	Access-Request id=0
3	244.621270	10.0.2.15	10.45.0.1	RADIUS	211	Access-Request id=0, Duplicate Request
4	250.630653	10.0.2.15	10.45.0.1	RADIUS	211	Access-Request id=0, Duplicate Request
5	262.638599	10.0.2.15	10.45.0.1	RADIUS	211	Access-Request id=0, Duplicate Request
6	286.843331	10.0.2.15	10.45.0.1	RADIUS	211	Access-Request id=1
7	289.846637	10.0.2.15	10.45.0.1	RADIUS	211	Access-Request id=1, Duplicate Request
8	295.851312	10.0.2.15	10.45.0.1	RADIUS	211	Access-Request id=1, Duplicate Request
9	307.856133	10.0.2.15	10.45.0.1	RADIUS	211	Access-Request id=1, Duplicate Request
10	316.902951	10.0.2.15	10.45.0.1	RADIUS	211	Access-Request id=2
11	319.906534	10.0.2.15	10.45.0.1	RADIUS	211	Access-Request id=2, Duplicate Request
12	325.918594	10.0.2.15	10.45.0.1	RADIUS	211	Access-Request id=2, Duplicate Request
13	361.942386	10.45.0.2	10.45.0.1	RADIUS	211	Access-Request id=3
14	361.952215	10.45.0.1	10.45.0.2	RADIUS	92	Access-Challenge id=3
15	362.007651	10.45.0.2	10.45.0.1	RADIUS	398	Access-Request id=4
16	362.020173	10.45.0.1	10.45.0.2	RADIUS	1096	Access-Challenge id=4
17	362.032062	10.45.0.2	10.45.0.1	RADIUS	214	Access-Request id=5
18	362.040807	10.45.0.1	10.45.0.2	RADIUS	1096	Access-Challenge id=5
19	362.052217	10.45.0.2	10.45.0.1	RADIUS	214	Access-Request id=6
20	362.060693	10.45.0.1	10.45.0.2	RADIUS	1081	Access-Challenge id=6

Capture at Core's "backhaul" interface



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::f57a:b826:80c9:af9b	ff02::2	ICMPv6	56	Router Solicitation from 00:00:00:00:00:00
2	294.419184	10.45.0.2	10.45.0.1	RADIUS	211	Access-Request id=3
3	294.426867	10.45.0.1	10.45.0.2	RADIUS	92	Access-Challenge id=3
4	294.484143	10.45.0.2	10.45.0.1	RADIUS	398	Access-Request id=4
5	294.494966	10.45.0.1	10.45.0.2	RADIUS	1096	Access-Challenge id=4
6	294.508453	10.45.0.2	10.45.0.1	RADIUS	214	Access-Request id=5
7	294.515289	10.45.0.1	10.45.0.2	RADIUS	1096	Access-Challenge id=5
8	294.528684	10.45.0.2	10.45.0.1	RADIUS	214	Access-Request id=6
9	294.535619	10.45.0.1	10.45.0.2	RADIUS	1081	Access-Challenge id=6
10	294.572645	10.45.0.2	10.45.0.1	IPv4	1396	Fragmented IP protocol (proto=UDP 17, off=0, ID=43b1) [Reassembled in #11]
11	294.572962	10.45.0.2	10.45.0.1	RADIUS	250	Access-Request id=7
12	294.580159	10.45.0.1	10.45.0.2	RADIUS	92	Access-Challenge id=7
13	294.593864	10.45.0.2	10.45.0.1	IPv4	1396	Fragmented IP protocol (proto=UDP 17, off=0, ID=43b2) [Reassembled in #14]
14	294.594039	10.45.0.2	10.45.0.1	RADIUS	153	Access-Request id=8

Figure 5.5: RADIUS traffic captured at `backhaul` channel via Wireshark

– Conversely, user plane traffic from NAUN3 devices (e.g., `iperf3` data in Listing 5.10 and ping traffic in Listings 5.7 and 5.8) is shown to originate from the `clients` DNN IP addresses (e.g., `10.46.0.2`, `10.46.0.3`) assigned to their respective PDU sessions. This demonstrates effective segregation between the authentication control plane and the user data plane.

While these observations do not constitute a formal penetration test or vulnerability assessment, they confirm that the fundamental security design principles, strong local authentication via EAP-TLS, segregation of control and user plane traffic via distinct DNNs, and abstraction of local NAUN3 device identities from the 5GC, are correctly implemented and operational within the test environment. The envisioned enhancement suggests a pathway to more integrated policy control if desired in future iterations.

5.5 Discussion and Analysis

The validation results presented in the preceding sections provide substantial evidence for the functional viability and integrity of the proposed framework for integrating Wi-Fi-only/NAUN3 devices into a 5G network. This discussion will interpret these findings in the context of the initial research objectives and requirements, compare the solution with standard 3GPP approaches, and acknowledge observed limitations.

5.5.1 Interpretation of Results and Effectiveness in Meeting Requirements

The primary goal was to devise a solution that allows NAUN3 devices, which lack 5G credentials, to securely access 5G network services with minimal impact on the devices themselves and the standard 5GC.

- **Local Authentication (Requirement Met) (Metric ID-1):** The successful EAP-TLS authentication demonstrates that NAUN3 devices can be securely authenticated at the edge by the 5G-RG before any 5G resources are committed. This meets the requirement for a robust local authentication mechanism.

- **Individual Device Handling and Proxy Identity (Requirement Met) (Metric ID-3, ID-4 and ID-5):** The core concept of using a dedicated PDU session as a proxy identity for each NAUN3 device was successfully implemented and validated. The UE logs and Listing 5.6 and 5.11 clearly show the dynamic creation of distinct PDU sessions on the `clients` DNN, each with a unique 5GC-assigned IP address, corresponding to each authenticated NAUN3 device. The `interceptor` logs (see Listings 5.3, 5.3 and 5.5) further detail the interceptor application's role in orchestrating these PDU session establishments via `nr-cli` upon EAP success.
- **Traffic Mapping and Isolation (Requirement Met) (Metric ID-6 and ID-7):** The `ping -R` tests (Listings 5.7 and 5.8) and `iperf3` results (Listings 5.10 showing distinct source IPs `10.46.0.2` and `10.46.0.3`) confirm that traffic from each NAUN3 device is correctly *NATted* and routed through its unique PDU session. The Wireshark capture in Figure 5.4 and the dynamic `iptables` rules in Listing 5.11 further substantiate that the 5G-RG effectively maps local device traffic to its designated 5G PDU session, ensuring traffic isolation.
- **Lifecycle Management (Requirement Met) (Metric ID-8):** The logs in Figures 5.12, 5.14 and 5.15 demonstrate that upon simulated NAUN3 device disconnection, the `interceptor` correctly triggers local deauthentication, PDU session termination, and cleanup of associated routing rules and DHCP permissions.
- **Traffic Segregation (Requirement Met) (Metric ID-7) :** The capture seen in Figure 5.5 confirm that authentication control plane traffic (RADIUS/EAP) is successfully segregated onto the `backhaul` DNN, using the 5G-RG's PDU session IP designated for this purpose, while NAUN3 user plane traffic utilizes the separate `clients` DNN PDU sessions.

On top of meting this metrics, minimal core network and device impact was achieved as the solution operates without requiring any modifications to the NAUN3 devices beyond standard EAP-TLS supplicant capabilities. The 5GC (Open5GS) interacts with the 5G-RG as a standard UE requesting PDU sessions; no changes to core NFs were needed beyond configuration (DNNs, subscriber data for the 5G-RG). This fulfills the critical requirement of minimal disruption.

A large contributor to this was the gateway-centric logic approached to the design. The specialized logic for NAUN3 authentication relay, PDU session orchestration, and traffic mapping is concentrated within the 5G-RG (`ue` VM), primarily within the custom `interceptor` application and configurations of `hostapd` and `dnsmasq`.

CHAPTER 6

Conclusion

Concluding the dissertation, this chapter synthesizes the research conducted, summarizing the key findings and contributions toward integrating Wi-Fi-only devices in 5G environments. Re-visits the problem statement and research objectives, discusses the implications of the results, acknowledges the limitations encountered during the study, and proposes potential avenues for future research and development in this domain.

6.1 Summary of Research and Key Findings

This research confronted the intricate problem of integrating Wi-Fi-only devices, which fit under NAUN3 devices as per 3GPP definition, into the 5G System. These devices, lacking native 5G credentials such as a USIM, are inherently unable to directly authenticate with or be managed by the 5GC through standard 5G procedures. The primary objectives were therefore to design a robust and secure local authentication mechanism for these devices, to propose an innovative identity management solution enabling the 5GC to handle their traffic without necessitating modifications to the devices or the core network NFs, and to validate this comprehensive framework within a realistic, albeit simulated, 5G environment.

The proposed solution is designed around the usage of a 5G-RG, which functions as an intelligent mediation point. The framework first mandates a local EAP-TLS authentication process for each NAUN3 device, where the 5G-RG acts as an EAP authenticator (or relay), forwarding authentication dialogues to an external, operator-controlled RADIUS server. Upon successful local authentication, the cornerstone of the solution is activated: the 5G-RG, orchestrated by a custom software component (the `interceptor`), requests a dedicated PDU Session from the 5GC specifically for that authenticated NAUN3 device. This PDU Session, established on a designated `clients` DNN, effectively serves as a dynamic "proxy identity" for the NAUN3 device within the 5G system. The `interceptor` is crucial in linking the outcome of the local authentication to the subsequent 5G PDU session management and the dynamic configuration of traffic routing rules. To maintain a clear separation of traffic, a

distinct `backhaul` DNN is utilized for the 5G-RG's own operational communications, such as the RADIUS messages exchanged with the EAP server.

The validation phase, conducted through a series of experiments in a simulated multi-VM environment, confirmed the framework's functional success. Key achievements include the consistent and secure EAP-TLS authentication of NAUN3 devices, as evidenced by detailed logs from all involved components (`wpa_supplicant`, extthostapd, FreeRADIUS, and the interceptor). Following authentication, the dynamic establishment of unique PDU sessions on the `clients` DNN for each NAUN3 device, along with the creation of corresponding `uesimtunX` network interfaces on the 5G-RG, was achieved. Both local (via DHCP managed by `dnsmasq`) and 5GC-assigned IP addresses were correctly allocated. End-to-end data connectivity was verified through `ping -R` tests and `iperf3` traffic generation, which also confirmed correct NAT operation via the per-device PDU session IPs. Traffic isolation between multiple NAUN3 devices, each utilizing its unique PDU session, was successfully demonstrated, supported by the dynamic `iptables` and policy routing rules. The system also effectively managed the lifecycle of these resources, cleaning up PDU sessions and associated configurations upon simulated device disconnections. Furthermore, control plane traffic (RADIUS/EAP) was shown to be correctly segregated onto the backhaul DNN, and the NAUN3 device's local identity remained concealed from the 5GC NFs. Repeated tests indicated an average onboarding delay of approximately 27 seconds.

6.2 Contributions of the Dissertation

This dissertation offers several significant contributions to the challenge of integrating diverse, non-5G-native devices into contemporary 5G networks.

The primary contribution is the design and proof-of-concept implementation of a practical, gateway-centric framework that enables Wi-Fi-only/NAUN3 devices, which lack standard 5G credentials, to securely access 5G network services. This solution is particularly relevant for scenarios where modifying end-user devices or the 5GC is not feasible.

A key innovation is the novel application of per-device PDU Sessions as a dynamic proxy identity mechanism. Managed by the 5G-RG, this approach allows the 5GC to handle traffic for individual NAUN3 devices without requiring these devices to have a SUPI or undergo direct 5GC authentication. Each PDU session acts as a distinct, manageable network presence for the corresponding NAUN3 device.

The research also demonstrates how local, strong authentication (EAP-TLS) can be tightly coupled with 5G session management procedures at the network edge (5G-RG). This ensures that only verified devices are granted access to 5G resources via these proxy PDU sessions, addressing a critical security consideration.

The development of a working proof-of-concept in a simulated 5G environment using open-source tools (Open5GS, UERANSIM) and custom orchestration logic also validates the architectural design. This implementation confirms that the proposed framework can operate with minimal impact on standard 5G NFs (requiring only configuration) and on the NAUN3 devices themselves (requiring only standard EAP supplicant capabilities).

Collectively, this work addresses a significant identity and authentication gap for a specific but increasingly common class of devices, offering a pathway for their managed integration into the 5G ecosystem, thereby enhancing the versatility and reach of 5G services.

6.2.1 Comparison with Standard 3GPP Methods and State of the Art

Standard 3GPP mechanisms for non-3GPP access typically involve the UE itself (or a function like N3IWF/TNGF) handling the interface to the 5GC. For devices behind an 5G-RG that are not 5G-capable (NAUN3), 3GPP TS 23.316 discusses "Connectivity Group IDs" [40] where groups of devices on a LAN segment can be mapped to a PDU session established by the 5G-RG. More recent Release 19 additions [33] [41] [43] introduce the "Non-3GPP Device Identifier" to allow for differentiated QoS for individual devices within a PDU session.

This project's solution aligns with the trend of providing more granular management for devices behind an 5G-RG. However, it offers a distinct approach:

- **Explicit Local Authentication as a Prerequisite:** Standard 3GPP methods [43] state that such non-3GPP devices are not directly authenticated by the 5GC. This project implements a robust EAP-TLS local authentication step as a prerequisite for any 5G resource allocation, a crucial aspect not explicitly detailed for NAUN3 devices in the standard flows for QoS differentiation.
- **PDU Session as Proxy Identity vs. QoS Tag:** While 3GPP R19 uses "Non-3GPP Device Identifiers" mainly for QoS differentiation within a PDU session, this project uses the dedicated PDU session itself as the primary proxy identity for the NAUN3 device in the 5G-RG. This provides a stronger isolation boundary and a direct handle for per-device policy and IP management by the 5G-RG. The 3GPP approach [41] can lead to a PDU Session Modification or Establishment for differentiated QoS, which is functionally similar in outcome for a single device but our approach makes this one-to-one mapping fundamental.

The implemented solution can be seen as a specific instantiation of how a 5G-RG could manage NAUN3 devices, extending the concept of Connectivity Groups to a per-device granularity and integrating a necessary local authentication layer.

6.3 Discussion of Limitations

While the validation phase confirmed the functional viability of the proposed framework, several limitations were identified, warranting consideration for future work and potential real-world deployments:

A notable limitation is the onboarding delay. Systematic testing revealed an average end-to-end delay of approximately 33.1634193 seconds (and 5.0078050 seconds of standard diviation) from the initiation of the `wpa_supplicant` on the NAUN3 device to its acquisition of a local IP address. This comprehensive duration includes the EAP-TLS handshake, all RADIUS communications, the PDU session establishment orchestrated by the interceptor (which relies on CLI command execution and polling for UERANSIM), and the final local DHCP lease

acquisition. While a significant improvement from some initial outlier measurements, and functional for a proof-of-concept, this delay could impact user experience in time-sensitive applications and represents a key area for optimization.

The data plane performance and scalability of the solution were confirmed at a functional level but not rigorously benchmarked. While `iperf3` tests demonstrated successful data transfer and isolation for a small number of concurrent devices, with observed bitrates suitable for many common applications, the study did not extend to stress testing under heavy load or with a large number of NAUN3 devices. The reliance of the `interceptor` application on executing system commands (`nr-cli`, `iptables`, `ip route/rule`) for each device's lifecycle events could potentially become a performance bottleneck on the 5G-RG at higher scales. A detailed analysis of CPU/memory impact on the 5G-RG under such conditions was also outside the scope of this work.

The use of NAT for traffic mapping from NAUN3 devices to their respective PDU Sessions, while effective for enabling outbound connectivity and providing a unique external IP presence per device, inherently introduces limitations typically associated with NAT. This primarily restricts the ease of initiating connections to the NAUN3 devices from external parties in the data network, as these devices are not directly addressable by their local IPs from the outside. Similarly, direct peer-to-peer communication between two NAUN3 devices, each behind its own NAT on the 5G-RG (if their traffic is routed externally and back), would require NAT traversal techniques or application-level relaying, which were not explored in this work.

Regarding security, while EAP-TLS provides strong local authentication for the NAUN3 devices and traffic segregation was demonstrated, the custom `interceptor` application itself and its control interfaces (e.g., the `hostapd` control socket) would necessitate further security hardening and thorough vulnerability assessment before any consideration for production deployment.

The complexity of dynamic traffic management on the 5G-RG is another consideration. The `interceptor`'s routing handler module successfully managed `iptables` and policy-based routing rules for the tested scenarios. However, ensuring conflict-free, secure, and performant rule management for a very large and highly dynamic set of devices would introduce significant operational complexity.

Finally, the attempt to integrate a physical 5G RedCap modem (Quectel RG500Q-GL) highlighted substantial physical hardware integration challenges. Issues related to proprietary drivers, kernel version dependencies, and a lack of comprehensive public documentation for advanced multi-PDU session features (like QMAP) made it difficult to replicate the fine-grained PDU session control achieved in the UERANSIM-based simulated environment. This underscores the potential gap between simulated proofs-of-concept and the practical deployment on diverse physical RG platforms, which would likely require considerable modem-specific adaptation and driver-level work.

6.4 Envisioned Enhancement

The findings and limitations of this research open several avenues for future investigation and development to enhance the proposed framework:

A primary focus should be on optimization of the onboarding delay. This could involve exploring more efficient mechanisms for PDU session control by the `interceptor`, such as direct API-based interactions with the 5G-RG's UE stack or modem if supported by future UERANSIM versions or different hardware platforms, thereby avoiding CLI parsing and polling. Optimizing RADIUS server response times and the EAP-TLS exchange, or even investigating the feasibility of pre-establishing a pool of `clients` DNN PDU sessions that can be rapidly assigned to newly authenticated devices, could also yield significant improvements.

Comprehensive performance and scalability analysis is essential. Future work should involve rigorous testing with a substantially larger number of concurrent NAUN3 devices under various traffic load conditions. This would help identify potential bottlenecks in the 5G-RG (CPU, memory), the `interceptor` application, or the 5GC, and quantify the framework's scalability limits. Exploring more performant traffic mapping mechanisms, such as Extended Berkeley Packet Filter (eBPF)-based solutions as an alternative to `iptables` for large-scale deployments, could also be beneficial.

Enhanced security hardening of the 5G-RG environment and the `interceptor` application is crucial. This includes securing the control interfaces used by the `interceptor`, implementing robust input validation, and potentially employing secure tunneling mechanisms like IPSec for RADIUS traffic over the backhaul PDU session, especially if it traverses untrusted network segments.

Further research could explore deeper integration with advanced 5G features. The per-device PDU session model provides a strong foundation for applying granular policies. Future work could investigate how this could be leveraged for fine-grained network slicing per NAUN3 device or dynamic QoS adjustments through integration with the 5GC's Policy Control Function (PCF). This could build upon the envisioned enhancement where the RADIUS server and 5G-RG securely communicate authenticated local identity information (EAP identity, MAC address) to the PCF/UDM, allowing the PCF to apply user-specific profiles to the correct proxy PDU session.

Expanding support for other authentication methods beyond EAP-TLS could increase the framework's applicability. This might include exploring MAC-based authentication for very simple IoT devices (with careful consideration of the associated security implications) or other EAP types suitable for different device capabilities and security requirements.

Addressing the limitations of NAT for inbound connections could also be explored, perhaps through integration with UPF capabilities for port forwarding or specific application-level gateways if required for certain NAUN3 device services. One potential avenue to mitigate NAT-related issues and improve inbound addressability for NAUN3 devices could be the exploration of *Framed Routing*. In this scenario, the RADIUS server, upon successful authentication, could provide *Framed-Route* attributes to the 5G-RG. These attributes would suggest static

routes to be installed on the 5G-RG, potentially assigning a routable IP address or a small subnet directly to the NAUN3 device's PDU session. This would require the 5G-RG to install these routes and ensure that the 5GC (specifically SMF/UPF) is also aware of how to route traffic for these Framed-IP-Addresses towards the correct 5G-RG and subsequently to the PDU session tunnel associated with the NAUN3 device. Such an approach could enable more direct inbound connectivity and facilitate peer-to-peer communication, though it would necessitate careful coordination of IP address management and routing policies between the RADIUS infrastructure, the 5G-RG, and the 5GC.

Addressing the challenges of physical modem integration remains a significant area. Continued efforts to work with diverse physical 5G modems and RG platforms, including deeper investigation into modem-specific APIs, QMAP functionalities, and driver development or adaptation, would be necessary for real-world deployment. Findings from such work could also inform potential contributions to standardization efforts if gaps in managing NAUN3-type devices via RGs persist.

Finally, investigating mobility scenarios, such as when the 5G-RG itself is mobile or when NAUN3 devices roam between different local access points (potentially managed by different 5G-RGs, would be a valuable extension to assess the framework's robustness and adaptability in more dynamic network conditions.

6.4.1 User-Specific QoS Policies

During the development of the current solution, another was envisioned for a possible future iteration. While the current implementation effectively conceals NAUN3 identities from the 5GC, a future enhancement could involve a secure communication channel between the RADIUS server and the 5GC (e.g., PCF/UDM). Upon successful EAP-TLS authentication, the RADIUS server could inform the 5GC about the authenticated EAP identity (which could be linked to a broader user or device profile known to the operator), the MAC address of the NAUN3 device, and the 5G-RG's identity (e.g., its SUPI or the IP of its `backhaul` PDU session from which the RADIUS request was relayed). To apply user-specific QoS, the 5GC would then need to query the 5G-RG (which maintains the MAC-to-PDU-session mapping) to identify which specific PDU session (established under the 5G-RG's SUPI) corresponds to the target NAUN3 device's MAC address. Once this correlation is made, the 5GC (specifically the PCF) could apply user-specific QoS policies to this now-identified PDU session, even if the NAUN3 device itself doesn't have a traditional IMSI-based subscription. This would allow for a richer, policy-driven service differentiation based on the authenticated local identity, bridging the local authentication domain with the 5GC's policy framework without exposing NAUN3 MAC addresses directly during PDU session establishment by the 5G-RG.

6.5 Final Concluding Remarks

This dissertation has successfully designed, implemented, and validated a novel gateway-centric framework for the secure integration of Wi-Fi-only/NAUN3 devices into 5G networks. By innovatively employing local EAP-TLS authentication as a prerequisite for the dynamic

establishment of per-device PDU sessions—which act as proxy identities within the 5GC, this research addresses a critical gap in current 5G architectures concerning the authentication and individualized management of devices lacking native 5G credentials. The validation results robustly demonstrate the functional correctness of this approach, including secure device onboarding, unique PDU session allocation, effective traffic isolation and mapping, and proper resource lifecycle management, all achieved with minimal impact on the standard 5GC and the NAUN3 end devices.

The primary takeaway from this work is that a 5G-RG, augmented with custom orchestration logic, can serve as an effective mediation point to bridge the local network domain of unauthenticated devices with the credential-based 5G system. This provides a practical pathway for extending 5G services to a broader range of devices. While acknowledging the identified limitations, such as the onboarding delay and the need for further scalability and performance testing, this research lays a solid foundation for future enhancements. The proposed solution not only offers a tangible approach to a current integration challenge but also aligns with the evolving 3GPP vision of more granular service control for devices in converged network environments. Ultimately, this work contributes to the broader goal of creating a more inclusive, flexible, and truly ubiquitous 5G ecosystem.

References

[1] F. Franciso, F. Miguel, and C. Rui, «Heading to a successful private digital convergence», Altice Labs, Tech. Rep., May 2022, p. 3.

[2] *System architecture for the 5g system (5gs)*, 3GPP TS 23.501 v19.2.0, 3rd Generation Partnership Project, Dec. 2024, p. 41.

[3] W. 5. W. Group, «5g and wi-fi ran convergence», Wireless Broadband Alliance, Tech. Rep., Apr. 2021, p. 59.

[4] *Security architecture and procedures for 5g system*, 3GPP TS 33.501 v19.0.0, 3rd Generation Partnership Project, Sep. 2024, p. 32.

[5] CableLabs, «A comparative introduction of 4g and 5g authentication», Cable Labs, Tech. Rep., 2019, p. 3.

[6] CableLabs, «A comparative introduction of 4g and 5g authentication», Cable Labs, Tech. Rep., 2019, p. 1.

[7] S. Eleftherakis, D. Giustiniano, and N. Kourtellis, «Sok: Evaluating 5g protocols against legacy and emerging privacy and security attacks», *arXiv (Cornell University)*, Sep. 2024. DOI: 10.48550/arxiv.2409.06360. [Online]. Available: http://arxiv.org/abs/2409.06360.

[8] Z. Li, W. Wang, C. Wilson, *et al.*, «Fbs-radar: Uncovering fake base stations at scale in the wild», in *Internet Society Symposium on Network and Distributed System Security (NDSS)*, Feb. 2017.

[9] S. B. Byeongdo Hong and Y. Kim, «Guti reallocation demystified: Cellular location tracking with changing temporary identifier», in *Internet Society Symposium on Network and Distributed System Security (NDSS)*, Feb. 2018.

[10] CableLabs, «A comparative introduction of 4g and 5g authentication», Cable Labs, Tech. Rep., 2019, p. 4.

[11] *Security architecture and procedures for 5g system*, 3GPP TS 33.501 v19.0.0, 3rd Generation Partnership Project, Sep. 2024, p. 30.

[12] *System architecture for the 5g system (5gs)*, 3GPP TS 23.501 v19.2.0, 3rd Generation Partnership Project, Dec. 2024, p. 48.

[13] *System architecture for the 5g system (5gs)*, 3GPP TS 23.501 v19.2.0, 3rd Generation Partnership Project, Dec. 2024, p. 538.

[14] *Security architecture and procedures for 5g system*, 3GPP TS 33.501 v19.0.0, 3rd Generation Partnership Project, Sep. 2024, p. 37.

[15] *Security architecture and procedures for 5g system*, 3GPP TS 33.501 v19.0.0, 3rd Generation Partnership Project, Sep. 2024, p. 46.

[16] *Security architecture and procedures for 5g system*, 3GPP TS 33.501 v19.0.0, 3rd Generation Partnership Project, Sep. 2024, p. 48.

[17] *Security architecture and procedures for 5g system*, 3GPP TS 33.501 v19.0.0, 3rd Generation Partnership Project, Sep. 2024, p. 52.

[18] *Security architecture and procedures for 5g system*, 3GPP TS 33.501 v19.0.0, 3rd Generation Partnership Project, Sep. 2024, p. 49.

[19] *Security architecture and procedures for 5g system*, 3GPP TS 33.501 v19.0.0, 3rd Generation Partnership Project, Sep. 2024, p. 232.

[20] *System architecture for the 5g system (5gs)*, 3GPP TS 23.501 v19.2.0, 3rd Generation Partnership Project, Dec. 2024, p. 243.

[21] *Numbering, addressing and identification*, 3GPP TS 23.003 v19.1.0, 3rd Generation Partnership Project, Dec. 2024, p. 20.

[22] *Numbering, addressing and identification*, 3GPP TS 23.003 v19.1.0, 3rd Generation Partnership Project, Dec. 2024, p. 21.

[23] *System architecture for the 5g system (5gs)*, 3GPP TS 23.501 v19.2.0, 3rd Generation Partnership Project, Dec. 2024, p. 244.

[24] *Numbering, addressing and identification*, 3GPP TS 23.003 v19.1.0, 3rd Generation Partnership Project, Dec. 2024, p. 29.

[25] W. 5. W. Group, «Private 5g and wifi convergence», Wireless Broadband Alliance, Tech. Rep., Apr. 2023, p. 3.

[26] *System architecture for the 5g system (5gs)*, 3GPP TS 23.501 v19.2.0, 3rd Generation Partnership Project, Dec. 2024, p. 57.

[27] *5g wireless wireline convergence architecture*, TR-470i2, Broadband Forum, Mar. 2022, p. 8.

[28] *System architecture for the 5g system (5gs)*, 3GPP TS 23.501 v19.2.0, 3rd Generation Partnership Project, Dec. 2024, p. 66.

[29] *System architecture for the 5g system (5gs)*, 3GPP TS 23.501 v19.2.0, 3rd Generation Partnership Project, Dec. 2024, p. 67.

[30] *Security architecture and procedures for 5g system*, 3GPP TS 33.501 v19.0.0, 3rd Generation Partnership Project, Sep. 2024, p. 128.

[31] *Security architecture and procedures for 5g system*, 3GPP TS 33.501 v19.0.0, 3rd Generation Partnership Project, Sep. 2024, p. 132.

[32] *Security architecture and procedures for 5g system*, 3GPP TS 33.501 v19.0.0, 3rd Generation Partnership Project, Sep. 2024, p. 139.

[33] *Wireless and wireline convergence access support for the 5g system*, 3GPP TS 23.316 v19.1.0, 3rd Generation Partnership Project, Mar. 2025, p. 29.

[34] *Wireless and wireline convergence access support for the 5g system*, 3GPP TS 33.501 v19.0.0, 3rd Generation Partnership Project, Dec. 2024, p. 37.

[35] *Wireless and wireline convergence access support for the 5g system*, 3GPP TS 33.501 v19.0.0, 3rd Generation Partnership Project, Dec. 2024, p. 41.

[36] *Security architecture and procedures for 5g system*, 3GPP TS 33.501 v19.0.0, 3rd Generation Partnership Project, Sep. 2024, p. 279.

[37] *Wireless and wireline convergence access support for the 5g system*, 3GPP TS 33.501 v19.0.0, 3rd Generation Partnership Project, Dec. 2024, p. 23.

[38] *Wireless and wireline convergence access support for the 5g system*, 3GPP TS 33.501 v19.0.0, 3rd Generation Partnership Project, Dec. 2024, p. 25.

[39] *Wireless and wireline convergence access support for the 5g system*, 3GPP TS 33.501 v19.0.0, 3rd Generation Partnership Project, Dec. 2024, p. 10.

[40] *Wireless and wireline convergence access support for the 5g system*, 3GPP TS 23.316 v18.6.0, 3rd Generation Partnership Project, Sep. 2024, p. 27.

[41] *Wireless and wireline convergence access support for the 5g system*, 3GPP TS 23.316 v19.1.0, 3rd Generation Partnership Project, Mar. 2025, p. 95.

[42] *Procedures for the 5g system*, 3GPP TS 23.502 v19.3.0, 3rd Generation Partnership Project, Mar. 2025, p. 447.

[43] *System architecture for the 5g system (5gs)*, 3GPP TS 23.501 v19.3.0, 3rd Generation Partnership Project, Mar. 2025, p. 564.

[44] *Policy and charging control framework for the 5g system*, 3GPP TS 23.503 v19.3.0, 3rd Generation Partnership Project, Mar. 2025, p. 110.

APPENDIX A

Key Differences between NAUN3 and N5GC devices

Table A.1: Key Differences between NAUN3 and N5GC devices

Feature	**NAUN3 Devices**	**N5GC Devices**
5G Capability	No 5G capability, cannot access 5GC directly.	Limited 5G capability, requires assistance to connect to 5GC.
Authentication	Local (e.g., Wi-Fi passphrase, PIN).	5GC authentication via EAP and W-AGF.
Access Type	Wireless (e.g., Wi-Fi via 5G-RG).	Wireline (e.g., fiber via W-AGF).
Subscription Records	None in UDM/UDR; operates under 5G-RG policies.	Unique subscription records separate from CRG.
NGAP Connections	Not applicable.	Separate NGAP connections per device.
Session Handling	Handled by the 5G-RG.	Handled by W-AGF and 5GC.
Purpose	Legacy IoT or low-capability devices.	Wireline devices requiring 5GC services.
Example	Smart home appliance using Wi-Fi.	Desktop computer on a fiber network.

APPENDIX B

Authentication for untrusted non-3GPP access

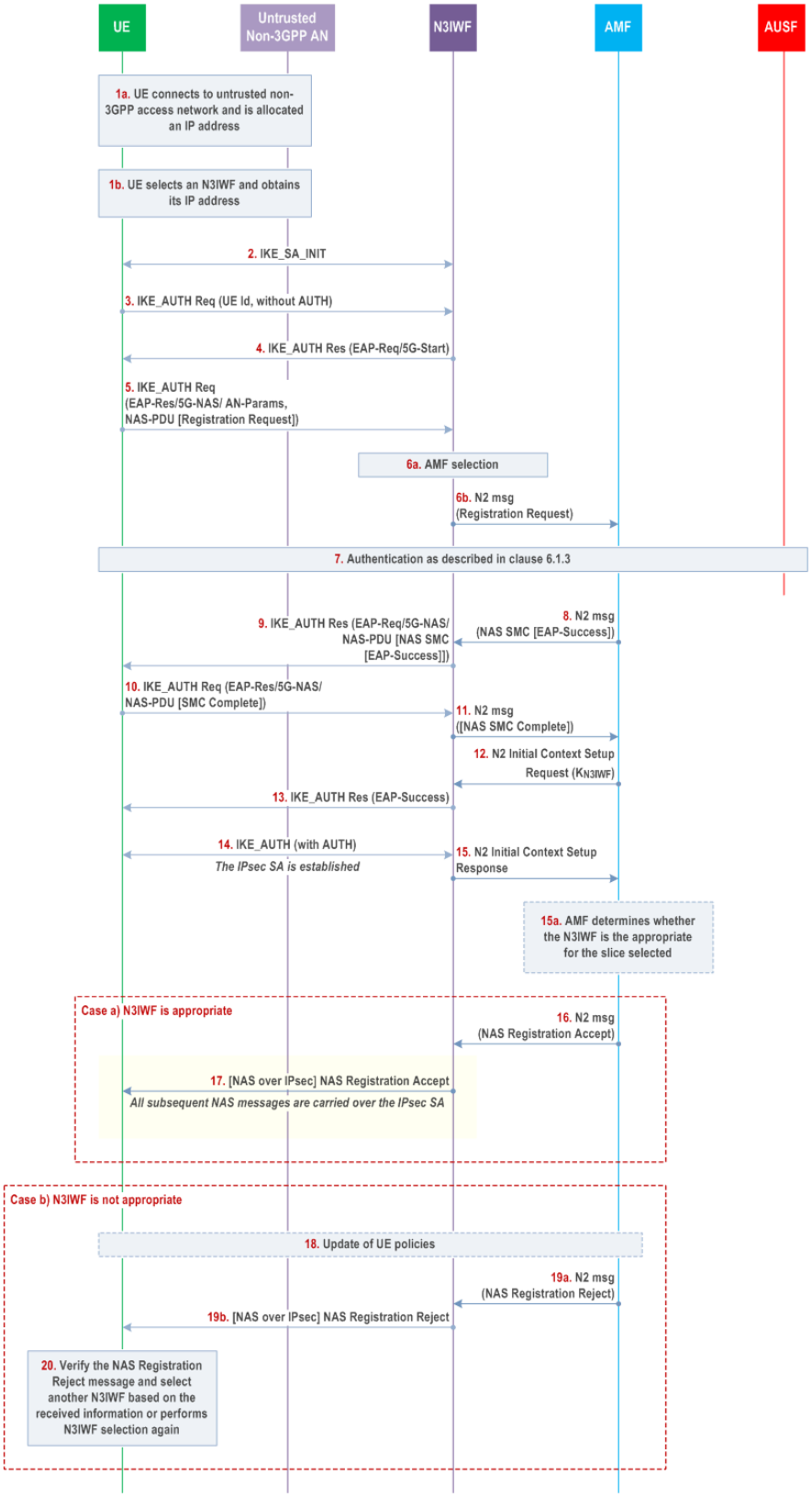


Figure B.1: Authentication for untrusted non-3GPP access

APPENDIX C

Authentication and PDU Session establishment for trusted non-3GPP access

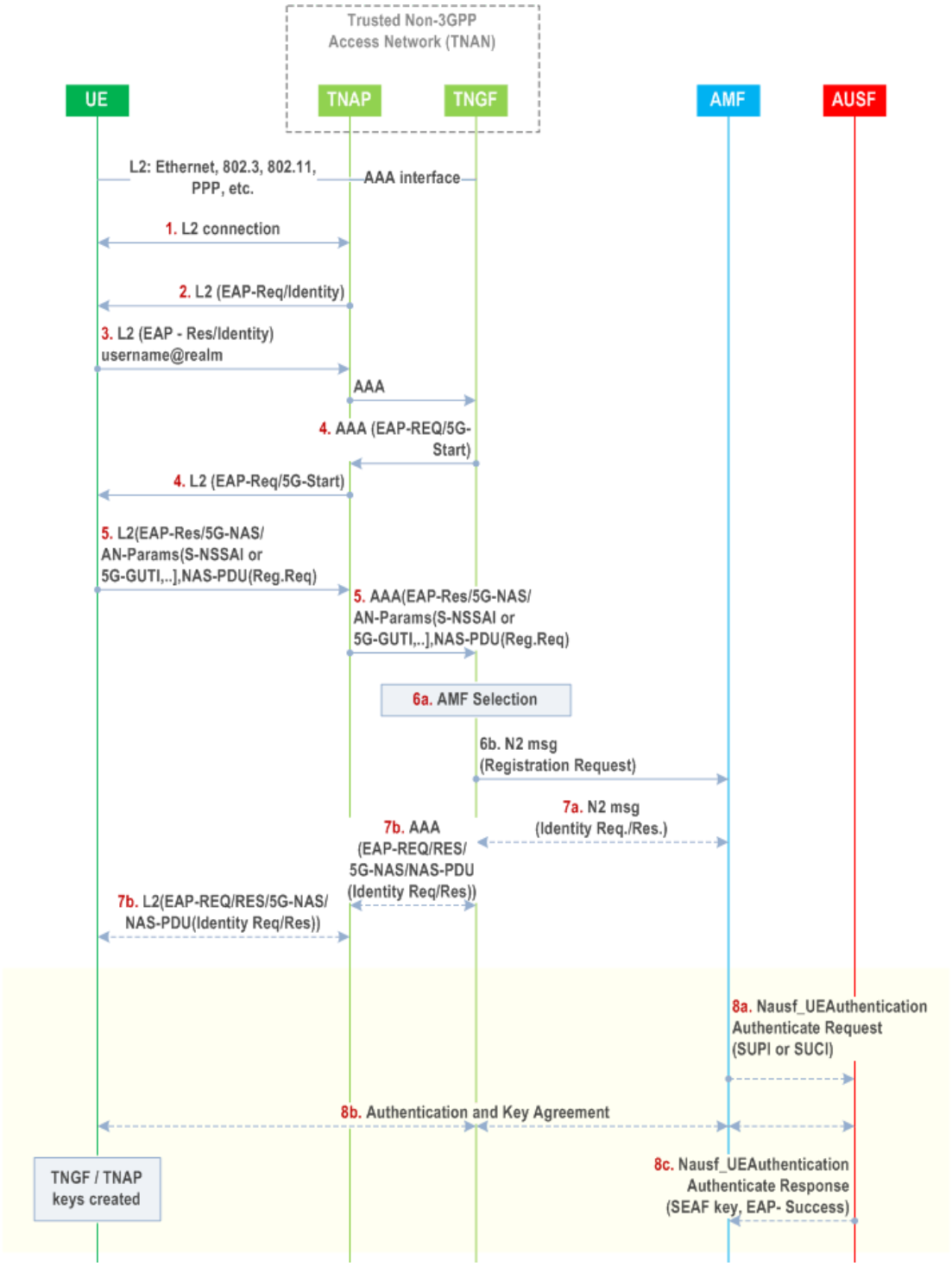


Figure C.1: Authentication and PDU Session establishment for trusted non-3GPP access

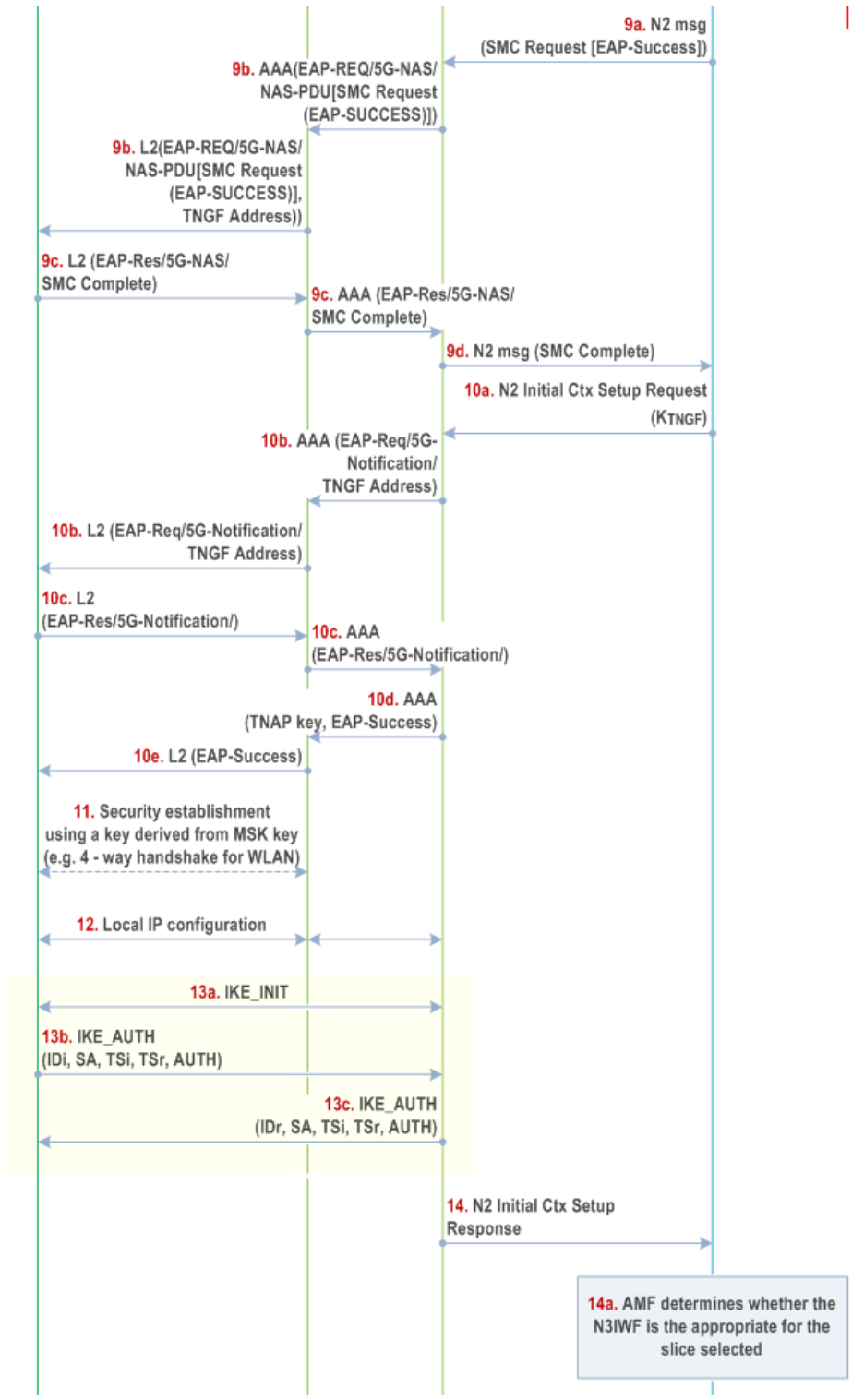


Figure C.2: Authentication and PDU Session establishment for trusted non-3GPP access (continuation)

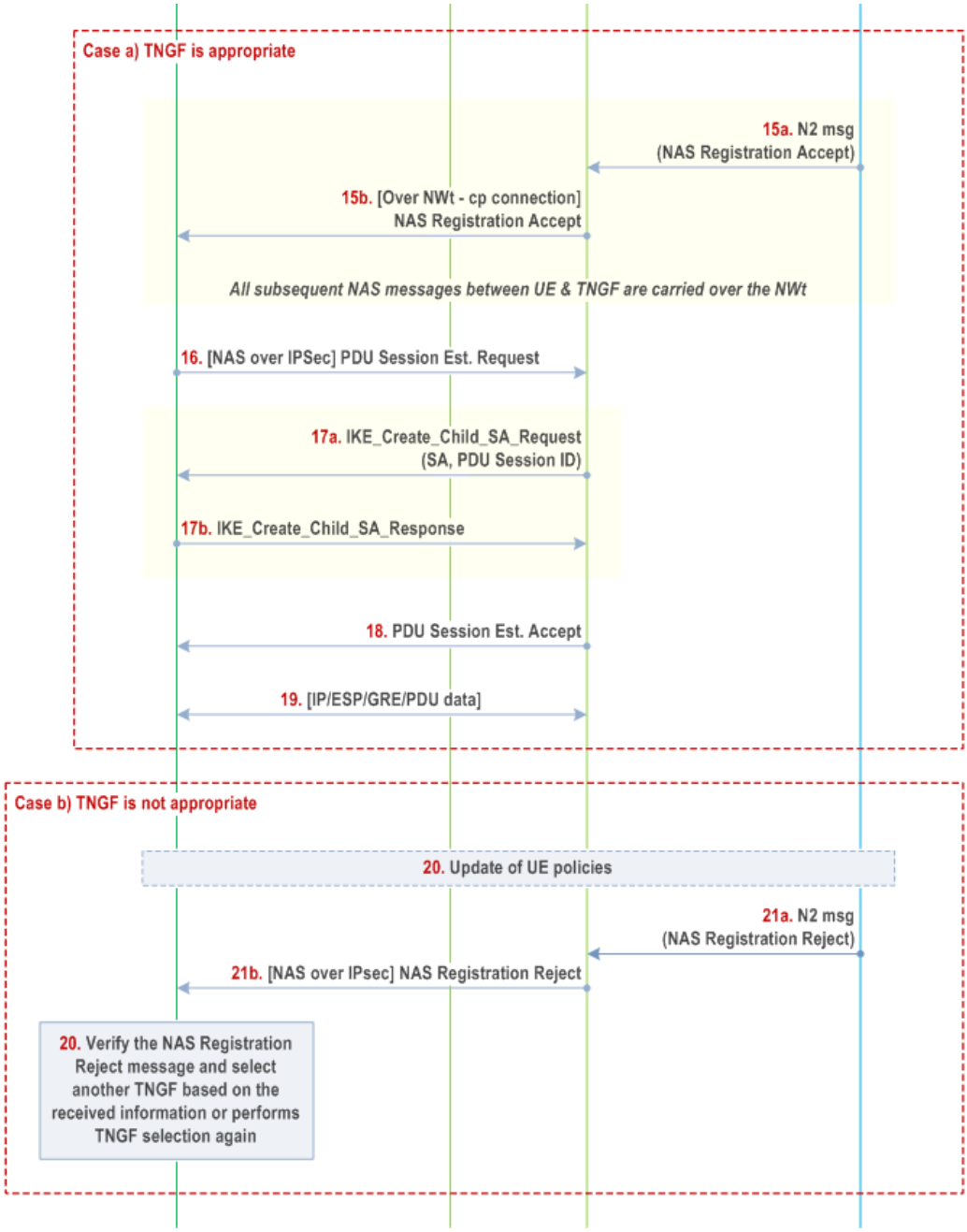


Figure C.3: Authentication and PDU Session establishment for trusted non-3GPP access (continuation)

APPENDIX D

Detailed registration and authentication flow of a N5GC device to the 5GC

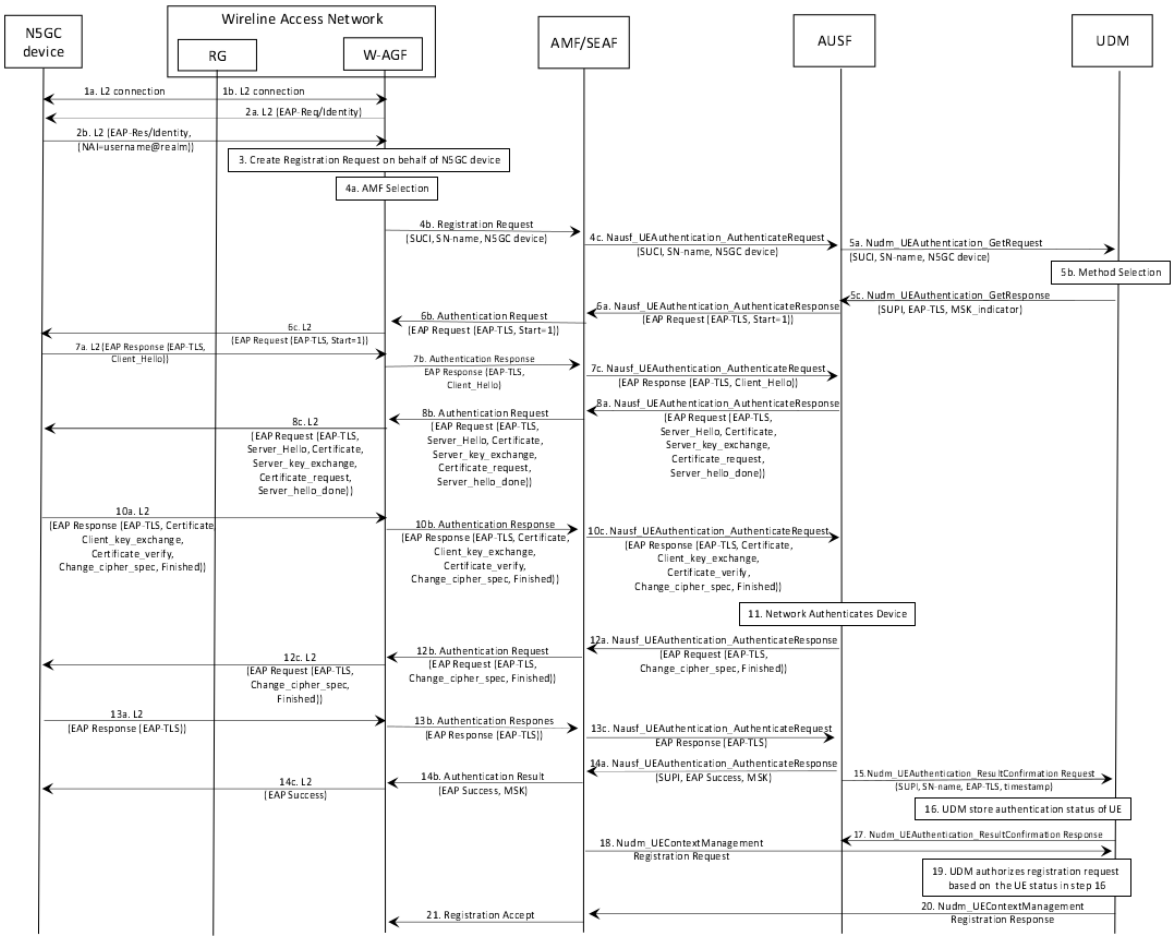


Figure D.1: Detailed registration and authentication flow of a N5GC device to the 5GC