TLS Performance Evaluation in the Control Plane of a 5G Core Network Slice

by

Alireza Hosseini Vasoukolaei

A thesis submitted to the
Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of the requirements for the degree of

Master of Information Technology

The School of Information Technology
Carleton University
Ottawa, Ontario
2021

©Copyright

Alireza Hosseini Vasoukolaei

Abstract

3GPP has recommended 5G service providers adopt Transport Layer Security (TLS) to secure communications among different Network Functions (NFs) in 5G core network slice. It is imperative to study the impact of TLS on 5G control plane traffic. Previous research has shown that TLS adds costs to protocols, like Session Independent Protocol (SIP) and The Onion Router (ToR). This study aims to study different TLS cipher suites impact on network performance parameters such as packet size, time, and the number of messages sent in distinct 5G control flows. We emulated a 5G core network slice using an open-source project and measured the overhead on the network parameters while 5G slice components transmit control plane traffic on both cases of TLS and no TLS. We examined the impact of different cipher suites of TLS namely, AES128-GCM-SHA256, AES256-GCM-SHA384, and 3DES-EDE-CBC-SHA with key exchange algorithms of RSA (Public Key Encryption) and Elliptic Curve Diffie Hellman Ephemeral (ECDHE) on the NF registration with Network Repository Function (NRF), UE registration, and PDU session establishment. We extended the analysis to TLS1.2 and TLS1.3 overhead using the same cipher suite (AES128-GCM-SHA256) on the NF registration procedure.

Acknowledgments

Throughout my thesis, I have received a great deal of support, guidance, and help.

Initially, I would like to thank my supervisor, Professor Ashraf Matrawy, whose directives, expertise, and knowledge sharing were imperative and invaluable in formulating the research questions and methodology. He guided me well through each stage of the thesis process.

I would like to acknowledge Dr. Danish Sattar for his wonderful collaboration. You helped me with automation techniques that helped me in my measurements.

My special thanks to my wife who provided me supports, wise counsel, and had a sympathetic ear. Finally, I would like to appreciate my parents who have always been there for me.

Table of Contents

Abstract	ii
Acknowledgments	iii
Table of Contents	iv
1 Introduction	3
1.1 Overview	3
1.2 Research Problem	5
1.3 Research Questions	5
1.4 Thesis Contributions	6
2 Background	8
2.1 5G Network Technology	8
2.1.1 5G Network Slicing	8
2.1.2 Lifecycle of a Network Slice Instance	10
2.1.3 Network Function Registration	11
2.1.4 User Equipment Initial Registration to a 5G Slice	13
2.1.5 PDU Session Establishment	15
2.2 TLS Record Protocol	17
2.2.1 TLS Record Protocol	19

2.2.2	TLS Handshake Protocol	20
2.2.3	Alert Protocol	21
2.2.4	Application data Protocol	23
3	Literature Review of The Related Works	24
3.1	TLS Impact on Protocols	24
3.2	5G Signalling Security and Protocols Selection	25
3.3	Technologies to Secure 5G Slicing	27
3.4	Identifying 5G Applications Sensitive to Latency	27
4	Experimental Setup and Methodology	29
4.1	Testbed	30
4.2	gNodeB and UE	31
4.3	Tool For Capturing and Analysis	31
4.4	Data Collection	32
4.5	Scenarios	32
4.5.1	TLS Overhead On Network Function Registration	32
4.5.2	TLS Processing Time Overhead On Initial Registration of a UE to 5G Network Slice	36
4.5.3	PDU Session Establishment Setup Time	37
4.5.4	Results Representation	38
5	Network Function Registration with Network Repository Function	40
5.1	Test with RSA key exchange	41
5.1.1	3DES-EDE-CBC-SHA	41
5.1.2	AES256-GCM-SHA384	43
5.1.3	AES128-GCM-SHA256	45

5.2	Test with ECDHE Key Exchange	47
5.2.1	3DES-EDE-CBC-SHA	48
5.2.2	AES256-GCM-SHA384	50
5.2.3	AES128-GCM-SHA256	52
5.3	Results Comparison	54
5.3.1	RSA	54
5.3.2	ECDHE	58
5.4	Test with TLS 1.3 RSA-AES128-GCM-SHA256	63
6	TLS Impact On The UE Initial Registration With a 5G Core Slice	69
6.1	Commissioning 5G Core, UE, gNodeB and Verifying UE Provisioning in UDM	70
6.2	UE Initial Registration with a 5G SBA Core Slice	70
6.3	Test with RSA key Exchange	72
6.3.1	3DES-EDE-CBC-SHA	72
6.3.2	AES256-GCM-SHA384	73
6.3.3	AES128-GCM-SHA256	73
7	TLS Impact On the UE PDU Session Establishment With a 5G Core Slice	75
7.1	Commissioning 5G Core, UE, gNodeB and Verifying UE Provisioning in UDM	76
7.2	UE PDU Session-Establishment with a 5G SBA Core Slice	77
7.3	Test with RSA key Exchange	78
7.3.1	3DES-EDE-CBC-SHA	79
7.3.2	AES256-GCM-SHA384	80
7.3.3	AES128-GCM-SHA256	80

8	Conclusions	83
8.1	Summary	83
	List of References	85
	Appendix A	94
A.1	Commissioning 5G SBA Slice and Verifying Control Plane Service Op- erations According to 3GPP	94
A.2	NF Registration With TLS1.2 Cipher Suites	96
A.3	Experiment with simultaneous launch of NFs	96

List of Figures

2.1	Simplified representation of lifecycle management of a network slice	11
2.2	Simplified representation of NF registration with NRF	12
2.3	Simplified representation of UE registration to a 5G core slice	14
2.4	Transport Layer Security Protocol	18
2.5	TLS Record Protocol	19
2.6	TLS Handshake Protocol	20
2.7	Simplified representation of TLS handshake procedure	22
2.8	Message format of TLS alert protocol	22
2.9	Message format of TLS application protocol	23
4.1	Each block represents a sub-process of the total registration process for both cases of TLS and no TLS.	33
4.2	Showing start and end measurement points with no TLS	34
4.3	Showing start and end measurement points with TLS	34
4.4	Simplified Representation of UE initial registration with a 5G Core Slice	37
4.5	Simplified Representation of PDU Session Establishment To a Network Slice	38
5.1	The comparison of total packet size overhead on the registration pro- cess with TLS cipher suites and RSA key exchange.	55

5.2	The comparison of the packet size overhead for registration-only sub-process with TLS cipher suites and RSA key exchange.	56
5.3	The comparison of the extra message overhead on the total registration process with TLS cipher suites and RSA key exchange.	56
5.4	The comparison of decrease in number of messages for the registration-only sub-process with TLS cipher suites and RSA key exchange. . . .	57
5.5	The comparison of the time overhead by the TLS cipher suites on the total registration process.	58
5.6	The comparison of total packet size overhead on the registration process with TLS cipher suites and ECDHE key exchange.	59
5.7	The comparison of the packet size overhead for registration-only sub-process with TLS cipher suites and ECDHE key exchange.	60
5.8	The comparison of the extra message overhead on the total registration process with TLS cipher suites and ECDHE key exchange.	61
5.9	The comparison of decrease in number of messages for the registration-only sub-process with TLS cipher suites and ECDHE key exchange. . .	61
5.10	The comparison of the time overhead by the TLS cipher suites with ECDHE key exchange on the total registration process.	62
5.11	The comparison of the extra messages TLS1.2 and TLS1.3 have on the NF registration process compared to when the same procedure occurs over TCP, no TLS case.	65
5.12	The comparison of decrease in number of the messages by TLS12 and TLS13 for the registration-only messages.	66
5.13	The comparison of total packet size overhead between TLS12 and TLS13.	66
5.14	The comparison of the total packet size overhead between TLS 12 and TLS13.	67

5.15	The comparison of the total time overhead the TLS12 and TLS13 add to the registration process.	67
6.1	Simplified Representation of a UE Registration to a 5G Core Slice and specifying the capturing points to calculate the UE initial registration time.	71
6.2	Time overhead of TLS cipher suites of 3DES-EDE-CBC-SHA, AES128-GCM-SHA256, and AES256-GCM-SHA384 on UE initial registration. UE registration with no TLS is considered as the baseline.	74
7.1	This is the topology of our testbed that includes slice components along with external entities such as UE, gNodeB, and service. Also it shows the control plane and user plane flows.	77
7.2	Simplified representation of the flows related to establishing a PDU session and GTP-U. It illustrates the start and end capturing points to calculate the PDU session Establishment time in TLS and no TLS scenarios.	79
7.3	Time overhead of TLS cipher suites of 3DES-EDE-CBC-SHA, AES128-GCM-SHA256, and AES256-GCM-SHA384 on UE PDU session establishment. UE PDU session establishment with no TLS is considered as the baseline.	82
A.1	Calling TLS cipher suites from crypto library of GoLang	95
A.2	There are significant discrepancies in total registration time when NFs register with NRF concurrently.	97
A.3	The total registration time when the NFs register with NRF one at a time.	98

List of Tables

5.1	List of the cipher suites.	41
5.2	Total packet size and registration-only size overhead by the 3DES-EDE-CBC-SHA cipher suite.	42
5.3	Total message count overhead added by the 3DES-EDE-CBC-SHA cipher suite.	42
5.4	The registration only message count between each network function and NRF with the 3DES-EDE-CBC-SHA cipher suite.	43
5.5	The total registration time overhead added by TLS-3DES-EDE-CBC-SHA on the registration process.	43
5.6	Total packet size and registration-only size overhead by the AES256-GCM-SHA384 cipher suite.	44
5.7	The sent messages overhead due to using TLS AES256-GCM-SHA384 cipher suite for whole registration process.	44
5.8	The sent messages overhead due to using TLS AES256-GCM-SHA384 cipher suite for the registration-only operation.	45
5.9	The time overhead of the TLS-AES256-GCM-SHA384 on the whole registration process.	45
5.10	The traffic overhead of AES128-GCM-SHA256 cipher suite for the whole registration process and the registration-only sub-process. . . .	46

5.11	The total message count overhead added by AES128-GCM-SHA256 cipher suite.	46
5.12	The decrease in the registration only message count between each network function and NRF with AES128-GCM-SHA256 cipher suite. . .	47
5.13	The total registration time overhead by AES128-GCM-SHA256 on the registration process.	47
5.14	The traffic overhead of ECDHE-3DES-EDE-CBC-SHA cipher suite for the total registration process and the registration-only packets. . . .	48
5.15	The sent messages overhead due to using TLS-ECDHE 3DES-EDE-CBC-SHA cipher suite for whole registration process.	49
5.16	The sent messages overhead due to using TLS-ECDHE 3DES-EDE-CBC-SHA cipher suite for the registration-only operation.	49
5.17	The total registration time overhead added by TLS-ECDHE 3DES-EDE-CBC-SHA to the whole registration process.	49
5.18	The traffic overhead of TLS with ECDHE-AES256-GCM-SHA384 cipher suite for overall registration process and the registration-only packets.	50
5.19	The message count overhead added to the overall registration process by TLS-ECDHE-AES256-GCM-SHA384 cipher suite.	51
5.20	The decrease in total number of messages in registration-only subprocess with ECDHE-AES256-GCM-SHA384 cipher suite.	51
5.21	The time overhead TLS-ECDHE AES256-GCM-SHA384 on the whole registration process.	51
5.22	The packet size overhead of TLS with ECDHE-AES128-GCM-SHA256 cipher suite for the total registration process and the registration-only packets.	52

5.23	The sent messages overhead due to using TLS-ECDHE-AES128-GCM-SHA256 cipher suite for the whole registration operation.	53
5.24	The sent messages overhead due to using TLS-ECDHE-AES128-GCM-SHA256 cipher suite for the registration-only operation.	53
5.25	The time overhead TLS-ECDHE AES128-GCM-SHA256 adds on the whole registration process.	53
5.26	Total packet size and registration-only size overhead by the TLS1.3-RSA-AES128-GCM-SHA256 cipher suite.	63
5.27	The sent messages overhead due to using TLS1.3-RSA-AES128-GCM-SHA256 cipher suite for whole registration process.	64
5.28	The sent messages overhead due to using TLS1.3-RSA-AES128-GCM-SHA256 cipher suite for the registration-only.	64
5.29	The time overhead TLS1.3-RSA-AES128-GCM-SHA256 adds to the overall registration process.	64
6.1	UE registration process for both scenarios of no TLS and TLS with the cipher suite of 3DES-EDE-CBC-SHA.	72
6.2	UE registration process for both scenarios of no TLS and TLS with the cipher suite of AES256-GCM-SHA384.	73
6.3	UE registration process for both scenarios of no TLS and TLS with the cipher suite of AES128-GCM-SHA256.	73
7.1	PDU session establishment process for both scenarios of no TLS and TLS with the cipher suite of 3DES-EDE-CBC-SHA.	80
7.2	PDU session establishment process for both scenarios of no TLS and TLS with the cipher suite of AES256-GCM-SHA384.	80

7.3	PDU session establishment process for both scenarios of no TLS and TLS with the cipher suite of AES128-GCM-SHA256	81
-----	--	----

Abbreviations

AMF	Access and Mobility Management Function
AUSF	Authentication Server Function
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
DES	Triple Data Encryption
ECDHE	Elliptic Curve Diffie Hellman Ephemeral
GCM	Galois/Counter Mode
NF	Network Function
NRF	Network Repository Function
NSSF	Network Slicing Selection Function
NAS	Non-Asynchronous Stratum
NFV	Network Function Virtualization
PCF	Policy Control Function
PDU	Protocol Data Unit
PDN	Packet Data Network
RSA	refers to the Public-key Encryption
SMF	Session Management Function
SBA	Service Based Architecture
SDN	Software-Defined Networking

SHA	Secure Hashing Algorithm
TLS	Transport Layer Security
TCP	Transmission Control Protocol
UDM	Unified Data Management
UDR	User Data Repository
UPF	User Plane Function
UE	User's Equipment

Chapter 1

Introduction

1.1 Overview

Mobile system technologies have been evolving over the past years, and the latest version, 5G, is emerging. Nowadays the incentives of 5G technology, which are mainly performance and quality of service over 4G network technology, have been broadcasted to the market. Serving with such high-quality demands requires having radical changes and upgrades not only to the Radio Access Network (RAN), but also to the core part of 5G. There are many use cases expected to be served by 5G are enhanced Mobile Broadband (eMBB), Massive Machine-type Communication (mMTC), and Ultra-Reliable Low Latency (URLL) services and communications [1]. These new vertical services create new requirements for 5G network technology. For instance, latency in URLL communications is expected to be less than 1ms, in some services the users should have a data rate of 1Gbps, and the network should be able to handle the population of 100000 User Equipment (UE) per square kilometer [2]. These services require a network with certain characteristics and features to facilitate the services' key parameters specifications such as delay, speed, and volume of connections. Forth generation (4G) network technology is unable to cater to such demands as it has been

designed to serve mobile UE [3]. Nevertheless, 5G network technology is expected to support a wide range of applications with certain and distinct service requirements and profiles. One technology that will help in meeting demands is network slicing [4].

The network slicing architecture utilizes the foundation of network virtualization technology [5]. Therefore, 5G technology adopts the virtualization notion to split its physical infrastructure into several virtual logical networks. They are called network slices. Each network slice is composed of several virtual network functions that collectively create an environment to serve a range of services of heterogeneous characteristics [6].

The technology of network slicing revolves around instantiating several separate logical mobile networks hosted atop the same physical infrastructure. A network slice consists of a group or groups of logical Network Functions (NF) that are independent and carry out their relevant tasks enabling a slice to deliver services according to different service level agreements. Therefore, network slicing is a technology that enables both multi-tenancy and service-tailored composition of mobile networks [7], [8].

Network slicing benefits from the technologies such as Network Function Virtualization (NFV) and Software-Defined Networking (SDN) that facilitate sharing by segregating backbone and physical computation resources from the application software and separating the data from the control plane. Besides, network slices need to be isolated to fulfill the security requirements of certain services [9]. Therefore, resilient security actions to preserve the separation among several services, slices, and network functions on a shared platform are a must.

Recently standardization bodies such as ETSI and 3GPP have presented concepts about 5G network slicing.

5G employs service-oriented architecture (SOA) [10] principles as the foundation

for 5G service-based architecture (SBA) for control plane (CP) signaling [11].

1.2 Research Problem

The realization of the network slicing paradigm in 5G technology, which is underpinned by NFV and SDN technologies, provides the operators more flexibility to deploy the slice functional components at distant locations.

The 3GPP standard organization recommends 5G network operators use Transport Layer Security (TLS) protocol to secure inter/intra-slice communications. TLS provides three security objectives including authentication, confidentiality, and message integrity [12].

While the TLS deployment in 5G is recommended by the standardization bodies, it is important to find out the costs of TLS cryptography operations on network performance metrics such as time/latency, packet size overhead, and the number of messages, compared to the case where there is no TLS.

1.3 Research Questions

This thesis addresses the following research questions:

1. How does TLS 1.2 (with different cipher suites) impact the core control plane operation of NF registration with network repository function (NRF)?
2. How does different the key exchange mechanisms, namely RSA and ECDHE, affect the network performance metrics, such as traffic overhead, time, and message count, for the control plane service operation?
3. What is the impact of TLS 1.3 with cipher suite of AES128-GCM-SHA256 on the control plane of NF with NRF?

4. How does TLS 1.2 (with different cipher suites) impact the 5G core control plane operations that are triggered by User Equipment (UE) such as initial UE registration and Packet Data Unit (PDU) session establishment?

1.4 Thesis Contributions

After reviewing the literature and related works in the field of measuring the impact of TLS on different control plane operations of a 5G core slice, we realized that none of the studies have demonstrated the effect of TLS adoption on a 5G intra-slice communication. Several studies have been conducted to examine TLS on some protocols such as Session Independent Protocol (SIP) for Voice over IP, and The Onion Router (ToR).

To the best of our knowledge, we are the first to evaluate how TLS imposes overhead to 5G control plane procedures.

We emulated a 5G network using an open-source project and tried to adopt slice standards [13]. Further, we followed the recommendation of the 3GPP standard documents [14, 15] and used Golang crypto library to deploy a specific TLS cipher suite to secure intra-slice communication. We note that the results are specific to the environment used and the crypto library of Golang used for this study. Also, changing the values of parameters (e.g HTTP/2 IdleTimeout) will result in different values.

Our major contributions are listed below:

- We examine the impact of different cipher suites of TLS namely, AES128-GCM-SHA128, AES256-GCM-SHA384, and 3DES-EDE-CBC-SHA with key exchange algorithms of RSA (Public Key Encryption) and ECDHE on the NF registration with NRF. The NF registration happens at the operation phase of

slice life cycle management. Chapter 5 includes the experiments and shows the results.

- The measurements are extended to test the newly defined TLS1.3 protocol with the cipher suite of AES128-GCM-SHA256 and show how much this cipher suite affects the NF registration. Chapter 5 includes the experiments and shows the results.
- We measured and evaluated the impact of each cipher suite on different key parameter indicators such as time, packet size, and the message count related to each 5G control plane scenario. Further, we demonstrated that the mode of operation of Cipher Block Chaining (CBC) is better with Galois/Counter Mode (GCM) in the experiments we performed.
- We showed how long it takes a 5G UE to make its initial registration with an operative 5G slice. We conducted the experiment when the slice components exchange application messages using different cipher suites. We conducted the tests for the case when a UE makes a PDU session establishment request that connects it to the service and data traffic. Chapters 6 and 7 demonstrate the results.

Chapter 2

Background

2.1 5G Network Technology

2.1.1 5G Network Slicing

Several upgrades and enhancements have been offered by the 5G network compared to the previous generation of mobile network technologies. The 5G follows a completely service-based architecture (SBA) that segregates the network control plane components from the user plane ones as described in detail in [16].

5G network can supply dynamic services that concurrently provide related quality of service requirements of each particular service applied by distinct industries [17]. As a result, the main standard bodies of 3GPP and ITU-R worked together on several use cases that at last, they all fall into the following three main categories: enhanced mobile broadband (eMBB) that aims high data volume applications (e.g. video-streaming), massive machine-type communications (mMTC), targeting low bit-rate devices (such as, IoT sensors and devices), and ultra-reliable low-latency communications (URLLC) for delay-sensitive services, such as, automatic vehicles, and health-care services [18].

Slicing a network infrastructure virtually can also be defined as dividing a physical network into multiple virtual ones that each one can be created according to a particular type of service application or end-user. Appreciating network function virtualization technologies, the shared hardware resources can be effectively and dynamically organized and assigned to virtual network slice instances according to the cloud users' requirement. In the context of 5G, a virtual slice consists of a set of network functional components, called network functions, that are tied up together for delivering a specific use case or business objective [19].

A typical network slice consists of three main layers [20]:

- **Services:** Specifics service requirements supported by network slice instances.
- **Network Slice Subnet Instance:** This encompasses the composition of both the physical and virtual network functions that build a network slice. It delivers the features requested by services.
- **Resources and Network Functions:** Supplies all the physical and logical compute nodes needed to commission a network slice.

There are multiple cases that have been examined in the field of network slicing in 5G networks. For instance, in the context of autonomous vehicles and eHealth services, a network slice requires to facilitate the services with URLLC. In [21], a range of medical data collection methods has been defined that emphasized how ultra-reliability and low latency of the slice operation can play a central role in the applications. However, there are applications like Content Distribution Network (CDN), in which network slices shall need other functional requirements such as high throughput and mobility features [22].

2.1.2 Lifecycle of a Network Slice Instance

5G networks employ the notion of NFV. The standard body of the European Telecommunications Institute (ETSI) has suggested a framework [23] specifying the lifecycle of a 5G network slice implemented on Network Function Virtualization Infrastructure (NFVI). The lifecycle management framework enables each network slice to constitute compositions, resources, and virtual topologies to fulfill the service key parameter and quality criterion [24]. In [8], the life cycle management of a network slice is thoroughly explained. It includes four main phases which are preparation and on-boarding, commissioning, activation, and decommissioning phase.

- **Preparation Phase:** This phase involves slice design, on-boarding, assessing the requirement for the slice creation, facilitating the network environment, and other necessary preparations required to be done before the creation of a network slice instance(NSI).
- **Commissioning Phase:** In the network slice commissioning phase, the required resources to set up different constituents of a network slice such as; Network Functions (VNF) and Physical Network Functions (PNF), created. The created components fulfill all the service level specifications realized in the preparation phase.
- **Run-time and Operation Phase:** A network slice can become active and operational after it passes the commissioning phase. At the operation phase, all the network functions start operating across a network slice to deliver the communication services. Network Slice instances are modified and or expanded due to topology changes and resources upgrade.
- **Decommissioning Phase:** At this phase, a network slice can be de-activated

after there is no need for the services it provides. Finally, when a network slice instance, including all its constituents, is no longer required, it is removed. It is called the decommissioning phase.

Figure 2.1 below depicts the phases concerned with slice lifecycle management:

Figure 2.1: Simplified representation of lifecycle management of a network slice

2.1.3 Network Function Registration

There is a network function within a slice named NRF. It preserves the profile of existing network functions and the service communications that they offer. A network function instance reveals its information to the related NRF when it primarily becomes active and operational or when a network function instance gets released and re-activated. This process occurs at the operation phase of slice life cycle [16].

The registration procedure uses a service operation called *Nnrf-NFManagement-NFRegister* to carry out the following actions:

1. To make NF registration with NRF by dispatching the NF profile that includes the parameters of NF.
2. To record the services associated with the NF.
3. To forward network discovery operation it needs to register NRF to another NRF.

Figure 2.2 illustrates how a network function registers its profile to NRF.

Figure 2.2: Simplified representation of NF registration with NRF

1. The NF initiates the registration process by sending a HTTP PUT message to NRF. Its application message corresponds to the URI naming the NF instance and the actual application traffic of the PUT message includes all the parameters that a network function needs to register. The NF identifier refer to by `nfInstanceId` is globally unique within a Public Land Mobile Network (PLMN) where the network function registers its information to the NRF. All the PLMN employ the unified `nfInstanceId` which is Universally Unique Identifier (UUID) version 4.

2. The NRF returns "201 Created" when it successfully makes the network function profile registration in its database. Furthermore, the NRF informs the registered NF about a "keep-alive timer" and the number of seconds they should wait between each of the hear-beat messages. If the NRF fails to register the attributes of the NF profile, then it sends back a "400 Bad Request" status code with a clear reason detailing the error.

It is noteworthy that [15] recommends that network functions shall establish a TLS connection to authenticate one another before they exchange any control plane messages such as network function profile registration.

2.1.4 User Equipment Initial Registration to a 5G Slice

A network slice enters the operation phase of the slice lifecycle when it becomes operative. At this stage, all the network functions have already registered their profile to NRF, so the slice is prepared to handle UE control plane messages to facilitate communication services. One of the messages relates to when a UE wants to connect to a 5G network slice. It undergoes the registration process to obtain access to the network [25]. There are different types of registrations as listed below [13]:

- **Initial Registration to a 5G core network:** It is triggered when a UE is switched on and wants to gain access to the network.
- **Periodic Location Update:** a UE restarts a registration process when it stays inactive for a while.
- **Mobility Registration Update:** It happens when a UE enters a new location area in both connected and idle mode and/or when a UE wants to update its parameters and capabilities.

- **Emergency Update.**

The general registration call process that includes initial registration involves all the registration procedures specified in [13] whereas periodic one does not require to process all the sequences used in other registration scenarios. Figure 2.3 depicts UE initial registration to a 5G network slice.

Figure 2.3: Simplified representation of UE registration to a 5G core slice

UE initiates the registration process by establishing Radio Resource Control (RRC) connection towards 5G base station (gNodeB) on SCTP (Stream Control Transmission Protocol). The registration request contains parameters, such as Registration type, 5G Global Unique Temporary Identifier (5G-GUTI), Last Tracking Area Identifier, Requested Network Slice Selection Assistance Information (NSSAI) that is used to uniquely identify a network slice, UE Capability, and List of PDU Sessions.

The following describes the flow in the figure2.3 briefly:

1. gNodeB selects the Access and Mobility Function (AMF) based on the received information from UE through SCTP. The gNodeB adds the "RAN UE Next Generation Application Protocol (NGAP) ID" and the "RRC Establishment Cause " to the message prior contacting to the selected AMF.
2. UE needs to be authenticated. So, the AMF checks the chooses Subscription Concealed Identifier (SUCI) and chooses the related Authentication server function (AUSF).
3. AMF triggers the authentication procedure towards AUSF. It aims to obtain the master key that AMF uses to extract NAS security keys and other security keys (s). The SUPI is also returned to the AMF.
4. For Registration in case of emergency and the one related to mobility, AMF contacts Session Management Function (SMF) to activate or re-activate PDU sessions requested to be activated.
5. As soon as AMF completes the registration, it returns the Registration Accept message to the UE. The UE replies with a registration complete message to AMF.

2.1.5 PDU Session Establishment

PDU session establishment is required when a 5G subscriber needs to set up a data plane connection. A user must register to a 5G slice before initiating the PDU session request. Upon establishing user plane connection, the user can use any subscribed communication services. PDU session establishments take place when any either of the following cases exists [26]:

- Set up a new PDU session.

- Migrate from a Packet Data Network (PDN) connection in 4G/LTE to PDU session in 5G network.
- Shifting an existing PDU session between non-3GPP access and 3GPP access.
- Request a PDU session for emergency services.

Below is a summary of the PDU session establishment procedure [13]:

1. gNode-B sends PDU session establishment request on behalf of the UE to AMF. The request is exchanged with NGAP UL NAS transport that includes parameters such as serving NSSAI, DNN, PDU session ID, request type and so on.
2. AMF checks the N1 message contents like the network slice selection identifier and the request type and then sends Nsmf-PDUSession-Create-SMContext-Request to SMF
3. According to the info provided by the UE, SMF communicates with Unified Data Management (UDM) and Policy Control Function (PCF) to collect the required info for PDU session creation.
4. Since it is an initial PDU session request SMF sends the related User Plane Function (UPF) an N4 Session Establishment Request.
5. UPF confirms the SMF request with N4 Session Establishment Response. The response includes the GPRS Tunnelling Protocol (GTP) information.
6. Upon receiving the last step, SMF ensures that the end point of the GTP tunnel is created and SMF sends AMF a Namf-Communication-N1N2MessageTransfer with Tunnel Details for N2 message and PDU session details in N1 Container

7. When AMF receives the N1N2 Message it sends gNode-B a NGAP PDU session Setup Request along with N2/N1 parameter from SMF.
8. At this point gNode-B setups a GTP tunnel with the UPF and send the N1 parameters to UE.
9. After successful creation of GTP tunnel, gNode-B sends back NGAP PDU session Setup Response to AMF .

2.2 TLS Record Protocol

Netscape managed to design and develop Secure Socket Layer (SSL) Protocol in 1994 that was purposefully to provide three main security requirements of confidentiality, integrity, and authenticity [27]. Back in the time when the Internet initially emerged, there were not many users using this infrastructure and in those days every user would trust one another. As a result, the data traveling among users was not needed to be encrypted and/or authenticated. As people have increasingly embraced using the internet since then, the necessity of encrypting the connection has become apparent. Netscape made a standard for encrypting communications between an Internet client and server as not any related security standards existed at that time. So SSL had extensively been deployed by diverse network operators and the Internet users. The initial version of SSL, SSL1.0, was not released to the public, however, SSL2.0 was released and published in 1995 [13]. There were several security flaws with SSL.2 and the SSL.3 was made to minimize them[12]. Finally, the Internet Engineering Task Force (IETF) stepped in and shoulder the responsibility of the security protocols post SSL version 3.0 and they changed its name to TLS. A new version of SSL 3.0 was developed and it was denoted as TLS 1.0 [28] . TLS 1.0 was backward compatible

and was able to inter-operate with SSL 3.0. The current version of TLS extensively used in the industry is TLS version 1.2 [12] .

TLS protocol operates mainly between two entities, namely a client and a server. It is supposed that there is a connection-oriented transport layer between those entities such as Transport Control Protocol (TCP). TLS can ensure the messages that are communicated between both parties are intact and detect any message interception and forgery [29] . TLS protocol is located between the transport layer and the application layer with the main duty of enforcing security measures. TLS consists of several sub-protocols that are listed below and shown in figure 2.4 [12] :

- TLS Record Protocol
- Handshake Protocol
- Alert Protocol
- ChangeCipher Spec Protocol, and
- Application Protocol

Figure 2.4: Transport Layer Security Protocol

2.2.1 TLS Record Protocol

TLS record protocol consists of different protocols. There are messages in each protocol that may contain several fields such as length, description, and content. This layer of TLS gets the data, that is subject to be transferred, and splits them into smaller and manageable data blocks, compresses the blocks if the compression is selected, digests the message with a MAC (Message Authentication Code) algorithm, ciphers it, and finally sends the encrypted data. At the other end, the received blocks are deciphered, verified whether it came from the right party, decompressed, and at last reassembled before being handed to serve by the higher protocol layer. The main duty of the TLS record protocol is to protect the integrity and confidentiality of an application message [12, 30, 31]. The figure 2.5 the TLS record structure.

Byte	0	1	2	3
0	Content Type	N/A		
1..4	TLS Version		Length	
	Major	Minor	Bits 15-8	bits 7-0
2	Protocol Message			
3	Message Authentication Code (MAC)			
4	Padding (Used in Block Ciphers)			

Figure 2.5: TLS Record Protocol
[31]

Several protocols utilize TLS record protocol including Handshake protocol, the alert protocol, the change cipher specification protocol, and the application data protocol. TLS record protocol operates in an environment named connection state where encompasses and determines a compression algorithm, an encryption algorithm, and a MAC algorithm. It is important to note that the Message MAC is always

computed and generated before encryption.

2.2.2 TLS Handshake Protocol

The handshake protocol as shown in figure 2.6 allows TLS entities, the client, and the server, to communicate messages of encryption and MAC algorithm, the keys needed for cryptographic purposes. It also enables the TLS entities to authenticate one another by employing the PKI (Public Key Infrastructure) mechanism. It is worth noting that no bulk encryption at the TLS record protocol would initiate until the handshaking successfully done [32] .

Byte	0	1	2	3
0	22	N/A		
1..4	TLS Version		Length	
	Major	Minor	Bits 15-8	bits 7-0
5..8	Message type	Handshake Message Data Length		
9	Handshake Message Data			

Figure 2.6: TLS Handshake Protocol
[31]

Figure 2.7 shows the series of messages that are exchanged between the client and the server during the handshaking operation [33] . In the figure message type can be one of the following messages:

- Client Hello Message: TLS client sends a hello message that contains TLS protocol version, session ID, cipher suite, compression method, and initial random numbers. After the client sent the hello message, it waits for the server hello message that contains the same parameters.
- Server Hello Message: the server sends back the chosen TLS version and the

cipher suite with a random number. It also sends its certificate that encompasses its public key. It also sends key exchange parameters along with its digital signature. After sending the server hello message is the server done message is sent to the client. It indicates the end of the server hello message along with its parameters.

- **Key Exchange Message:** At this point, the client receives all the parameters mainly including the server certificate and the server key parameters. The client needs to send its certificate if the server already requested that. the client needs to send it key parameters.
- **Client Change Cipher Spec Message:** With this message the client acknowledges it has all the information that required to begin the encryption so the next message from it will be ciphered.
- **Client Finished message:** it includes an encrypted summary of all the messages initiated from the client so far.
- **Server Change Cipher Spec and Finished Message:** The server acts the same as the client in previous message. This finished message aims to identify if there has been any man-in-the-middle attack during the handshake protocol process.

2.2.3 Alert Protocol

When there is an issue or failure between TLS entities, TLS alert protocol's role comes in to the picture. It is responsible to transmit the level of the failure and its description to the participating entities. TLS protocol brings down the connection if a level of fatal alert occurs [12]. In such a scenario, TLS protocol allows the other connections related to the same session to carry on, however it invalidates the session identifier

Figure 2.7: Simplified representation of TLS handshake procedure

to be used to establish any further subsequent connections. Figure 2.8 denotes the alert protocol.

Byte	0	1	2	3
0	21	N/A		
1..4	TLS Version		Length	
	Major	Minor	Bits 15-8	bits 7-0
2	Level	Description	N/A	
3	MAC (Message Authentication Code)			
4	Padding (Used only in block cipher)			

Figure 2.8: Message format of TLS alert protocol
[31]

2.2.4 Application data Protocol

The actual user traffic is called application data that are divided into smaller blocks of data, that is called fragments, then the optionally compressed, and finally are encrypted by the record layer protocol [12].

Byte	0	1	2	3
0	23	N/A		
1..4	TLS Version		Length	
	Major	Minor	Bits 15-8	bits 7-0
2	Level	Description	N/A	
3	MAC (Message Authentication Code)			
4	Padding (Used only in block cipher)			

Figure 2.9: Message format of TLS application protocol
[31]

Chapter 3

Literature Review of The Related Works

4G technology has been extensively commissioned and quite mature around the world, and researchers started paying attention to 5G. 5G mobile technology supplies more services and upgraded characteristics [34]. We can observe several trends clarifying demands for new services. For instance, ITU (International Telecommunication Union) provides a summary of three main application categories, containing eMBB, URLLC, and mMTC services [35]. To facilitate the new demands, 5G needs to make some changes in design and adopt new technologies for the RAN and the Core Network (CN). 5G architecture may deploy millimeter wave, beam-forming, massive-Multiple Input Multiple Output (MIMO). The 5G core utilizes SND, NFV, and network slicing to create a flexible network.

3.1 TLS Impact on Protocols

Securing Voice over Internet Protocol (VoIP) is examined in [36]. SIP caters to the signaling part of VoIP protocol and security it is crucial [37]. Similar to other Internet applications, SIP might be vulnerable to certain threats and attacks, namely social engineering threats, traffic attacks, Denial of Service (DoS), and service abuse [38–40].

Therefore, the SIP standard and specification recommend that SIP operators employ TLS as a standard approach to secure SIP signaling. Considering TLS performance cost on SIP, [36] ran several experiments to evaluate the amount of processing affect the TLS can have on the CPU of both the SIP client and server. The evaluation output mainly relies on if and how often the client and server establish a TLS connection as TLS session negotiation has a high cost of RSA (Rivest - Shamir-Adleman) public-key operation. In conclusion, they define a measurement-centric cost model that SIP operators may use in supplying the SIP server.

3.2 5G Signalling Security and Protocols Selection

Innovation and evolution of mobile communication protocols, which are mainly related to authentication, authorization, and update location, in 2/3G, 4G, and 5G has been discussed in [41] . 2/3G use MAP protocol and 4G uses Diameter [42] and for 5G according to the protocol examinations TCP/TLS/HTTP2/JSON, UDP/QUIC/HTTP2/JSON, and SCTP/Diameter, TCP/TLS/HTTP2/JSON is eventually selected as a protocol suite for 5G [41]. This protocol suite fosters great compatibility when it comes to none-telecommunication applications. While the paper elaborates HTTP/2 provides eases use of operator owned application functions and inter-working with third parties' applications, the new mobile technology, new use cases, and new blueprint will unwittingly create new issues to security protection and HTTP/2 can not be exempted. It is expected to have new threats since there have never been such a large number of usage scenarios in mobile communication systems, such as vehicle-to-vehicle, vehicle-to-infrastructure, health cases, industrial automation, smart cities, smart homes, and so on [43].

Article [44] emphasizes how protocol selection is a key in SBA style of 5G since

all the network functions communication is facilitate by SBA. It challenges the Diameter protocol that has been widely used in EPC (Evolved Packet Communication). It works well in mobile operator's network and has a deficit in working and being supported by external interfaces. The Diameter would not be a competent candidate due to it heavy operation and inflexibility. Therefore, enumerates a range of protocol candidates that can be used in a different layer of the protocol stack in SBA style of 5G core network technology. For instance, TCP, QUIC, and SCTP for the transport layer, HTTP1.1 and HTTP2 for the application layer, and REST and RPC for the API design style. Different performance parameters for the application layer protocol candidates are evaluated. Page Load Time (PLT) has been a metric to evaluate the performance of HTTP1.1 and HTTP2. They measured two parameters, one is the total amount of requests that are completed correctly and the number of requests that are done in a second. at last, They noticed how the efficiency of header compression, that is a new capability developed in HTTP2, lessened the bandwidth latency. They demonstrate that HTTP2 works better than HTTP1.1 by noticing the growth rate of HTTP2 PLT is less than HTTP1.1 as the network delay increases. They checked both application protocol's performance under high concurrency. They send 600,000 concurrent requests. the results reveal that HTTP2 has a greater success rate and higher throughput than HTTP1.1 as HTTP2 adopts multiplexing and does not need to establish a new TCP connection that includes a three-way handshake. The new header compression feature increased the request rate up to around 85 percent that makes HTTP2 a good fit for the 5G network where a lot of identical and replicated signaling messages exist. However, they and leave the evaluation of the transport for further research and studies.

3.3 Technologies to Secure 5G Slicing

The concept of network slicing has been elaborated in [45] that allows new characteristics of flexibility and further enhances security since it enables the building of several separated logical networks over the same shared hardware resources [34, 46]. Further, it explained the other concepts concerned in 5G network slicing, namely virtual and horizontal network slicing, and End-to-End network slicing [47–49]. They emphasized that the central expectation of 5G network slicing is isolation. A slice can be considered as a separated group of resources commissioned via the network orchestration which supplies a defined set of functions and services. [45] has come up with several isolation properties post examining 5G network slicing security issues [50]. They enumerated a list of technologies that assure the isolation problems [51]. One of the named technologies is based on VPN-based network slice isolation that employs special protocols such as IPSec, SSL/TLS (Secure Socket Layer/Transport Layer Security, DTLS (Datagram Transport Layer Security). With having slice isolation safety issues, properties, and listing the technologies to separate them, finally, they concluded the paper by delving into the related concerns to the future generation of telecommunications standards like the fifth generation [52–54].

3.4 Identifying 5G Applications Sensitive to Latency

Since the 5G network is expected to cater to a wide range of services as well as bringing changes to business models, there will no longer be a one-security-solution for all the entities. Although all the security measures in 4G have been taken into account for 5G, the new evolution in 5G architecture brings several threats to its main entities

that are examined in [55] . It further details some of the key services that 5G going to supply, namely, Vehicle to Everything (V2X), and Internet of Thing (IoT) [56–58] . It is emphasized how low latency is central in network slicing of 5G to ensure such new services can operate in new key parameter performance. Security requirement of the V2X has been elaborated with consideration of standard documents of [59, 60].

The paper extracted the security requirements defined by 3GPP and mentioned the standards of [61–63] can apply to different operational mechanisms in V2X such as Security for broadcast between UEs, public-key cryptography and long term certificates for authorization and accountability, and employing Pseudonymous Mobile Subscriber ID (PMSI) or encrypted International Mobile Subscriber Identity (IMSI) for privacy. However, it does not discuss the security between network functions within a Network Domain Security (NDS).

Chapter 4

Experimental Setup and Methodology

The thesis aims to examine the effect of TLS implementation on different control plane operations of a 5G core network slice. It further intends to quantify the TLS impact and overheads on the processes. To achieve that, we reviewed the related works to figure out the metrics that can be affected by the deployment of TLS. Granerud in [64] investigated TLS traffic generated by ToR and compared it with the common TLS traffic. The author assessed performance metrics such as the number of packets, round trip time, and latency, the cipher algorithm for both scenarios of TLS and without TLS. The studies in [36] evaluated the cost of TLS operation on SIP. The paper measured CPU utilization in different cases for instance: TCP without authentication and no encryption, TCP with authentication and encryption, TCP with mutual authentication. Author of [65] provided an overview and analysis of overhead caused by TLS. This paper looked into how much processing, latency, and traffic overhead TLS adds.

Considering and given the metrics used in different researches, we decided to use time, packet size, and the number of messages TLS adds in the context of 5G intra-slice communication. Towards satisfying these objectives, we need to emulate a 5G core network slice that encompasses all the default and specific network functions

including AMF, AUSF, NSSF, PCF, SMF, UDM, UDR, and UPF. Furthermore, some scenarios involve UE and gNodeB to connect to an active and operative slice via AMF. So, we emulated them as well to explore and test the control plane flows, particularly when a UE attaches to and requests for services through a 5G core slice.

The rest of this chapter explains the open-source projects and tools we used to emulate the testing environments and make our analysis. Also, it elaborates on the control plane scenarios of the 5G core slice in which we conduct the experiments and calculate the TLS performance overheads for the selected metrics as mentioned earlier.

4.1 Testbed

The experiments are conducted on Linux based operating system running on a Virtual Machine (VM) at Carleton University. To set up a 5G core network with all its related elements by SBA model, we chose the Free5GC stage II release ¹, which is an open-source project developed for 5G core mobile network technology. It is based on the fifth mobile service-based interface model and has been used in recent researches related to the fifth generation technology [66–70]. Free5GC incorporates the default and specific network functions including, AMF, SMF, NRF, NSSF, UDM, UDR, and AUSF. The testbed supports the N1 interface between AMF and UE. UE uses this interface to make a connection to a 5G network slice for making initial registration, PDU session establishment, and handover. N2 interface between gNodeB and AMF bridges the control plane signaling between UE and AMF. It also supports 5G-NGAP, 5G Non-Access Stratum (NAS), and 5G authentication that are layered and operate atop of SCTP protocol. SCTP is a transport-layer protocol that makes sure the

¹<https://www.free5gc.org/installations/stage-2-standalone-5g/>

information is exchanged reliably and in order. 5G network uses this protocol between gNodeB and AMF.

4.2 gNodeB and UE

We selected `gnbsim`² that is an open-source tool to emulate a standalone UE and gNodeB. It is developed according to the 5G mobile technology release 16. It uses the N1 and N2 interfaces to connect UE and gNodeB to the AMF, respectively. It is aimed to allow the tool's users to conduct different control plane tests initiated from the UE and gNodeB side such as initial registration and PDU session establishment. We run a light-weight python server to act as a compute node connected to the UPF of the Free5GC 5G core slice. It allows us to ensure the PDU session establishment reaches the point where the GTP(GPRS Tunneling Protocol) tunnel is established.

4.3 Tool For Capturing and Analysis

We selected Wireshark and `tcpdump` for tapping packets and for analysis purposes. Wireshark and `tcpdump` are open-source tools that are used for capturing traffic. Since we commissioned the testbed on a virtual machine and it tends to have several virtual interfaces, the capturing tools let us explore the list of the interfaces and choose the ones that facilitate and terminate the gNodeB, the UE, and the network functions as well.

²<https://github.com/hhorai/gnbsim>

4.4 Data Collection

We deploy different TLS cipher suites to allow the slice components to authenticate each other and secure their communications. We need to measure the control plane KPIs³ for scenarios in which the slice is running with TLS and the slice is running without TLS. We repeat the experiments twenty-five times and capture the related control plane traffic with tcpdump to assess different measurements such as TLS registration time, TLS traffic overhead, and the number of messages exchanged when a network function registers its profile with NRF. We use bash and python scripts to automate the experiments and overheads calculation. The automation aims to enable us to perform multiple experiments and minimize human errors for the calculations.

4.5 Scenarios

4.5.1 TLS Overhead On Network Function Registration

There are several overheads such as latency, additional traffic, and the number of messages TLS can impose on the network function registration process. The registration process occurs in the operation phase of a slice life-cycle management. At this phase, all the network functions follow the REST architecture to send their profile to the network repository function. The network repository function receives the network functions registration call and acknowledges the registration by sending the REST message to the related network function. We follow 3GPP [14] that recommends 5G network operators to employ TLS version 1.2 to secure connections and message exchanging between network functions within a network slice. An NF initiates the registration process with a TCP handshake in both considered scenarios.

³KPI means the same as Key Performance Indicator

In the no TLS case, NF establishes TCP connection by making a three-way TCP handshake before sending HTTP traffic. Afterward, it starts sending HTTP traffic according to [13]. The registration process is terminated when NRF sends back the confirmation. Figure 4.1 breaks down the complete registration scenario into different sub-processes including, TCP handshake, TLS handshake, and registration. It shows that TLS entities should handshake, depicted in block B, prior to performing any cryptography operations.

Figure 4.1: Each block represents a sub-process of the total registration process for both cases of TLS and no TLS.

TLS Time

To calculate how long it takes an NF to register with NRF when TLS is in action, We refer to the figures 4.2 and 4.3 to measure as follows:

In no TLS case as shown the figure 4.2, the start and end time capture points are specified.

In TLS case as shown in the figure 4.3, a network function makes a three-way TCP handshake before doing TLS handshake. Afterward, it starts sending HTTPS traffic according to the 3GPP registration procedure. The registration process terminates

when the network repository sends the confirmation. Capture start and endpoints are specified in the figure.

Figure 4.2: Showing start and end measurement points with no TLS

Figure 4.3: Showing start and end measurement points with TLS

Referring to Figures 4.2 and 4.3, Eq. 4.1 shows how we calculate TLS Time Overhead where:

- $T1$ is the time duration that NF takes to make a three-way TCP handshake with NRF.
- $T2$ is the time duration of exchanging HTTP traffic between NF and NRF.
- $T3$ is time duration NF and NRF require to make two-round trips TLS handshake prior ciphering the traffic.
- $T4$ is the time duration of exchanging HTTPS traffic between NF and NRF.
- T is the total time duration of NF registration with no TLS.
- Ts is the total time duration of NF registration with TLS.

$$T = T1 + T2$$

$$Ts = T1 + T3 + T4 \quad (4.1)$$

$$TLS \text{ Time Overhead} = Ts - T$$

TLS Packet Size Overhead

Referring to figure, 4.1 we work out how much traffic overhead each sub-process of NF registration with NRF adds when TLS1.2 is employed with different cipher suites. Eq. 4.2 shows how the TLS traffic overhead is calculated.

- $TCPs$ Accumulative bytes exchanged in TCP process, as shown in figure4.1 block A or D.
- $TLSs$ Accumulative bytes exchanged in TLS handshake process, figure4.1 block B.
- $TLSrs$ Accumulative bytes exchanged in registration process when TLS is active, figure4.1 block C.

- *NoTLSrs* Accumulative bytes exchanged in registration process when TLS is not active, figure 4.1 block E.
- *NoTLSts* Total traffic when there is no TLS.
- *TLSts* Total traffic when there is TLS.

$$NoTLSts = TCPs + NoTLSrs$$

$$TLSts = TCPs + TLSs + TLSrs \quad (4.2)$$

$$Total\ TLS\ Traffic\ Overhead = TLSts - NoTLSts$$

4.5.2 TLS Processing Time Overhead On Initial Registration of a UE to 5G Network Slice

Another 5G core control plane process we want to study is when a user equipment wants to attach to a 5G network slice. The UE has to go through a set of procedures to obtain access to the network and to receive its related services. This process is called UE initial registration. In 4G and LTE, the registration procedure takes place in parallel with PDN connection which is different compared to the 2G and 3G networks. The 5G is designed based on the notion of separating the user plane from the data plane; thus, both processes, the initial registration and PDU session establishment, do not happen concurrently.

Figure 4.4, that is perceived from 3GPP [13], simplifies a UE initial registration procedure.

We perform the UE initial registration in the case of having no secure connection of TLS among network functions. Then We capture the NGAP that carries the initial

Figure 4.4: Simplified Representation of UE initial registration with a 5G Core Slice

request that is sent from UE/gNodeB towards AMF. We capture the UL (Uplink) NAS packet that is sent when the registration procedure is completed. We can figure out the delta time from the subtracting time of UE registration acceptance from the NGAP message. We follow the same sequence when we enable TLS authentication and encryption among network components within a slice.

4.5.3 PDU Session Establishment Setup Time

We evaluate PDU session setup time to find out how long it takes a UE to have its initial data plane traffic connected towards certain services and/or internet [71]. We run the test to connect to a specific service accessible by the UPF from the N6 interface. We conduct the test in two different scenarios, one when the network slice components exchange control plane messages with no TLS and another one when they exchange the messages with TLS.

In each scenario, we look into the control plane messages related to PDU session request that leads to GTP tunnel establishment. We calculate the PDU session setup time is by taking the time gNodeB sends PDU session request to AMF and the

time gNodeB returns PDU session setup response to AMF. Afterward, the GPRS Tunneling Protocol (GTP) is established and the initial data plane traffic is routed between the UE and service. Figure 4.5 illustrates the procedure.

Figure 4.5: Simplified Representation of PDU Session Establishment To a Network Slice

[71]

4.5.4 Results Representation

We collect measurements that are absolute values in units of milliseconds for time, counts for the number of messages, and kilobytes for packer size overhead. We decided to present overhead made by TLS to no TLS case in percentage as well as the nominal values. It aims to provide a more practical evaluation. I applied the following equation to produce results in percentage:

Assume that $noTLS$ = Absolute value of metrics such as time, packet size, and number of messages, in no TLS case and TLS = Absolute value of the same metrics in TLS case, so the overhead for any metric is worked out by following equation:

$$TLS\ Overhead = \left(\frac{TLS - noTLS}{noTLS} \right) * 100$$

This is notable that the above formula is generic and applied for all the metrics.

Chapter 5

Network Function Registration with Network Repository Function

The realization of the slicing technology in the 5G core network distinguishes it from its former generations such as 4G, 3G, and 2G. To overcome the security issues, 3GPP recommends deploying TLS and mTLS for intra-slice and inter-slice communications, respectively [14]. It aims to ensure the messages exchanged among NFs are authentic, intact, and remain confidential. The effect of the TLS operations on 5G control plane flows is important to study. The goal of this chapter is to measure the impact of TLS1.2 performance on one of the key control plane procedures of a 5G core slice that is NF profile registration with NRF. We created a 5G core slice using an open-source project, Free5GC. Also, we configured the slice components to exchange messages with different TLS1.2 cipher suites. Finally, we calculated the impact of different cipher suites on the metrics of time, messages count, and packet size for NF registration with the NRF. Besides examining the TLS1.2, we looked into TLS1.3 as well and measured its impact on the NF registration process. The experimental setup is explained in details in appendix A.

5.1 Test with RSA key exchange

This section shows the tests using three cipher suites of TLS1.2 as shown in Table 5.1.

We note that the three cipher suites use RSA key exchange method.

Cipher Suite Name	Number of Iterations per NF
TLS-RSA-3DES-EDE-CBC-SHA	25
TLS-RSA-AES256-GCM-SHA384	25
TLS-RSA-AES128-GCM-SHA256	25

Table 5.1: List of the cipher suites.

5.1.1 3DES-EDE-CBC-SHA

The experiment aims to have the network functions of a 5G network slice transmit the application messages over a specific transport layer security cipher suite which is TLS12 RSA WITH 3DES-EDE-CBC-SHA.

The script repeated the process for all network functions one at a time. We used a python script to make the measurements to minimize human errors.

The following tables show the results for different metrics.

Table 5.2 depicts the absolute packet size values in KiloBytes for cases of no TLS and TLS with the 3DES-EDE-CBC-SHA cipher suite. Also, it shows the traffic overhead of the registration-only sub-process that assess how much the TLS cipher suite adds extra packet overhead for the application traffic.

Table 5.3 shows that slice components exchange more messages when they communicate their profile with NRF over TLS-RSA-With-3DES-EDE-CBC-SHA. The total number of messages overhead ranges around 20% among network functions compared to no TLS case. Notably, the extra messages are attributed to TLS handshake.

Packet Size							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Packet size-NoTLS(KBytes)	4.93	2.58	3.20	5.71	3.72	4.87	2.47
Packet size-TLS(KBytes)	7.22	4.85	5.52	8.04	6.03	7.13	4.79
Total TLS Packets Pver-head(%)	47%	88%	72%	41%	62%	47%	94%
Registration-only Packets Overhead(%)	5%	9%	9%	5%	7%	5%	12%

Table 5.2: Total packet size and registration-only size overhead by the 3DES-EDE-CBC-SHA cipher suite.

Total Number of Message							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Messages-NoTLS(Count)	18.76	17.28	18.48	18.72	18.6	19.28	16.72
Messages-TLS(Count)	22.45	21.28	22.38	22.91	22.33	22.46	21.44
TLS Message Overhead(%)	20%	23%	21%	22%	20%	16%	28%

Table 5.3: Total message count overhead added by the 3DES-EDE-CBC-SHA cipher suite.

Nonetheless, it is noteworthy that there is less number of messages transmitted in the registration sub-process. The drop ranges from 9% to 17% among different NFs as shown in Table 5.4. It is mainly caused by HTTP2 SETTING and WINDOW-UPDATE messages when TLS is active.

Table 5.5 reveals the TLS time overhead for NF to NRF registration when the cipher suite of 3DES-EDE-CBC-SHA is deployed. It shows the actual nominal values and the TLs time overhead. Almost all the network functions have to spend around 60% to 70% more time to complete the registration with NRF.

Number of Registration-Only Message							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Messages-NoTLS(Count)	15.76	14.28	15.48	15.72	15.6	16.28	13.72
Messages-TLS(Count)	13.44	12.26	13.37	13.89	13.32	13.44	12.43
Registration-only Message Overhead(%)	-15%	-14%	-14%	-12%	-15%	-17%	-9%

Table 5.4: The registration only message count between each network function and NRF with the 3DES-EDE-CBC-SHA cipher suite.

Total Registration Time							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Time-NoTLS(ms)	24.98	27.88	23.76	27.10	22.09	31.21	24.44
Time-TLS(ms)	41.12	46.71	39.39	44.79	38.43	45.70	42.97
Total Time Overhead(%)	65%	68%	66%	65%	74%	46%	76%

Table 5.5: The total registration time overhead added by TLS-3DES-EDE-CBC-SHA on the registration process.

5.1.2 AES256-GCM-SHA384

In this section, we experimented NF registration with NRF with the cipher suite of TLS-RSA-WITH-AES256-GCM-SHA384. We set up the slice with this specific cipher suite the same way we did for 3DES cipher suite.

Table 5.6 depicts the packet size in KiloBytes for different sub-processes of the registration process. Also, It shows the traffic overhead of the AES256-GCM-SHA384 cipher suite for the total registration process (TCP handshake + TLS handshake + Registration) and the registration only packets.

Table 5.7 shows the total number of messages sent across in NF registration with NRF for both cases of having the TLS cipher suite and no TLS. The table shows the nominal values and the messages sent overhead in percentage for all the NFs. Results show that TLS AES256-GCM-SHA384 increases the message count around 20% than

Packet Size							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Packet Size-NoTLS(KBytes)	4.93	2.56	3.21	5.73	3.70	4.83	2.50
Packet Size-TLS(KBytes)	7.11	4.74	5.41	7.93	5.90	7.06	4.69
Total TLS Packets Overhead(%)	44%	85%	68%	38%	60%	46%	88%
Registration-only Packets Overhead(%)	3%	6%	6%	3%	5%	4%	7%

Table 5.6: Total packet size and registration-only size overhead by the AES256-GCM-SHA384 cipher suite.

no TLS scenario.

Total Number of Message							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Messages-NoTLS(Count)	18.84	17.12	18.56	19.04	18.2	18.64	17.16
Messages-TLS(Count)	22.30	21.04	22.38	22.70	22.25	22.86	21.33
TLS Message Overhead(%)	18%	23%	21%	19%	22%	23%	24%

Table 5.7: The sent messages overhead due to using TLS AES256-GCM-SHA384 cipher suite for whole registration process.

Table 5.8 reports the same message overhead percentage but for the registration-only operation. The TLS AES256-GCM-SHA384 cipher suite decreased the number of registration only packet by around 11% tp 16% across the NFs. The drop is mainly due to the SETTING and Windows UPDATES messages of HTTP2 protocol.

Table 5.9 shows the impact of AES256-GCM-SHA384 on total registration time. Results show that the profile registration time overhead increases by a range of 50% to 60%.

Number of Registration-only Message							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Messages-NoTLS(Count)	15.84	14.12	15.56	16.04	15.2	15.64	14.16
Messages-TLS(Count)	13.28	12.03	13.37	13.69	13.24	13.85	12.27
Registration-only Message Overhead(%)	-16%	-15%	-14%	-15%	-13%	-11%	-13%

Table 5.8: The sent messages overhead due to using TLS AES256-GCM-SHA384 cipher suite for the registration-only operation.

Total Registration Time							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Time-NoTLS(ms)	22.23	21.41	23.06	23.37	26.27	27.75	20.76
Time-TLS(ms)	34.01	33.78	36.82	41.41	47.61	41.22	34.72
Total Time Overhead(%)	53%	58%	60%	77%	81%	49%	67%

Table 5.9: The time overhead of the TLS-AES256-GCM-SHA384 on the whole registration process.

5.1.3 AES128-GCM-SHA256

We tested the same key exchange scheme, RSA, with the cipher suite of AES128-GCM-SHA256.

Table 5.10 illustrates how much Kilobytes this cipher suite adds to the whole registration process. The absolute values encompass the total amount of the packets related to the whole NF's registration procedure including TCP handshake, TLS handshake, and registration. The extra packets are mainly related to the TLS handshake packet size exchanged in TLS handshake protocol. The table also specifies the extra kilobytes overhead realized in the registration-only part that ranges from 3% to 7% across all the network function.

Regarding the number of messages, table 5.11 depicts that overall the TLS AES128-GCM-SHA256 cipher suite brings around 20% more number of messages

Packet Size							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Packet Size-NoTLS(KBytes)	4.94	2.55	3.22	5.72	3.68	4.81	2.50
Packet Size-TLS(KBytes)	7.13	4.74	5.42	7.92	5.89	7.05	4.70
Total TLS Packet Overhead(%)	44%	86%	68%	38%	60%	47%	88%
Registration-only Packets Overhead(%)	3%	7%	5%	3%	5%	5%	7%

Table 5.10: The traffic overhead of AES128-GCM-SHA256 cipher suite for the whole registration process and the registration-only sub-process.

than no TLS case that is mainly attributed to the extra messages that TLS entities deal with TLS handshake protocol. However, the number of registration-only mes-

Total Number of Message							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Messages-NoTLS(Count)	18.88	16.88	18.76	19.00	18.00	18.40	17.16
Messages-TLS(Count)	22.50	21.18	22.59	22.60	22.00	22.64	21.41
TLS Message Overhead(%)	19%	25%	20%	19%	22%	23%	25%

Table 5.11: The total message count overhead added by AES128-GCM-SHA256 cipher suite.

sages sent back and forth is less than no TLS. The decrease accounts in range of 11% to 24% across the slice components as shown in Table 5.12.

Table 5.13 has the time overhead that AES128-GCM-SHA256 cipher suite has on the process. It carries both the relative nominal values and the time impact in the percentage. The total registration time overhead ranges from 30% to 50% among distinct NFs.

Number of Registration-only Message							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Messages-NoTLS(Count)	15.88	13.88	15.76	16	15	15.4	14.16
Messages-TLS(Count)	13.50	12.16	13.58	13.59	12.99	13.64	12.40
Registration-only Message Overhead(%)	-15%	-12%	-24%	-15%	-13%	-11%	-12%

Table 5.12: The decrease in the registration only message count between each network function and NRF with AES128-GCM-SHA256 cipher suite.

Total Registration Time							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Time-NoTLS(ms)	25.97	26.42	27.87	26.77	15.14	24.75	23.98
Time-TLS(ms)	37.43	38.82	36.32	40.14	23.67	33.79	35.82
Total Time Overhead(%)	44%	47%	30%	50%	56%	37%	49%

Table 5.13: The total registration time overhead by AES128-GCM-SHA256 on the registration process.

5.2 Test with ECDHE Key Exchange

This section aims to evaluate the TLS impact with ECDHE key exchange method that is based on Diffie Hellman mathematical equations. In this key exchange algorithm, there are three keys, two of which are private and one is public. The assumption is not to send the private key over the wire. Essentially the combination of the three keys gives us the secret key that is used for performing cryptography operations namely, encryption, decryption, and hash function, in TLS record protocol. The following subsections present the results of the experiments with TLS ECDHE with different cipher suites of 3DES-EDE-CBC-SHA, AES-256-GCM-SHA-384, and AES-128-GCM-SHA-256. The experiments are repeated for 25 times.

5.2.1 3DES-EDE-CBC-SHA

We measured the impact TLS 1.2 cipher suite of 3DES with the key exchange mechanism of ECDHE.

Table 5.14 shows the results of measurement for the total number of Kilobyte added to the registration procedure of different NFs. The total packet-size overhead of this TLS cipher suites compared to no TLS are presented for all the network functions. The packet size overhead for the total registration process mainly stems from exchanging extra messages of TLS handshake.

There is a traffic overhead added to the registration-only process that is coming from the TLS header. It ranges from 4% to 11% for the different network functions.

Packet Size							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Packet Size-NoTLS(KBytes)	4.89	2.56	3.19	5.70	3.70	4.82	2.50
Packet Size-TLS(KBytes)	7.32	4.94	5.62	8.06	6.13	7.24	4.86
Total TLS Packet Overhead(%)	50%	93%	76%	41%	66%	50%	94%
Registration-only Packets Overhead(%)	6%	11%	10%	4%	9%	6%	10%

Table 5.14: The traffic overhead of ECDHE-3DES-EDE-CBC-SHA cipher suite for the total registration process and the registration-only packets.

There are more messages when NFs talk to NRF over TLS-ECDHE 3DES-EDE-CBC-SHA in the total registration process compared to the same scenario in no TLS. The extra messages ranges from 19% to 24% for different components in the 5G core slice, table 5.15.

Nonetheless, table 5.16 there are fewer messages sent back and forth for the registration-only messages that result from the nature of HTTP2 protocol. The decrease ranges from 10% to 16% than no TLS scenario.

Total Number of Message							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Messages-NoTLS(Count)	18.20	17.00	18.32	18.64	18.28	18.48	17.24
Messages-TLS(Count)	22.61	21.44	22.65	22.23	22.70	22.70	21.30
TLS Message Overhead(%)	22%	26%	24%	19%	24%	23%	24%

Table 5.15: The sent messages overhead due to using TLS-ECDHE 3DES-EDE-CBC-SHA cipher suite for whole registration process.

Number of Registration-only Message							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Messages-NoTLS(Count)	15.20	14.00	15.32	15.64	15.28	15.48	14.24
Messages-TLS(Count)	13.61	12.43	13.64	13.21	13.69	13.69	12.29
Registration-only Message Overhead(%)	-10%	-11%	-11%	-16%	-10%	-12%	-14%

Table 5.16: The sent messages overhead due to using TLS-ECDHE 3DES-EDE-CBC-SHA cipher suite for the registration-only operation.

Table 5.17 shows the amount of time in milliseconds each network function spent to complete their profile registration. In TLS case each network function spends more time than no TLS that is mainly due to doing the handshake as well as performing encryption, decryption, and the hash function. The time overhead ranges around 60% to 70% for the majority of the network functions.

Total Registration Time							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Time-NoTLS(ms)	27.91	27.23	26.18	20.14	21.87	25.99	31.52
Time-TLS(ms)	46.34	45.80	41.87	35.52	39.21	42.81	46.32
Total Time Overhead(%)	66%	68%	60%	76%	79%	65%	47%

Table 5.17: The total registration time overhead added by TLS-ECDHE 3DES-EDE-CBC-SHA to the whole registration process.

5.2.2 AES256-GCM-SHA384

This aims to calculate the impact of AES-256-GCM-SHA384 with the ECDHE handshake algorithm. We followed the same commissioning procedure as demonstrated for ECDHE 3DES-EDE-CBC-SHA cipher suite and verified the slice is operational.

Table 5.18 shows the amount of the packets associated with the total registration process for both AES256-SHA384 cipher suite with ECDHE key exchange and no TLS. The total packet size overhead is mainly attributed to TLS handshake traffic. We note that the registration only packet overhead ranges from 3% to 8%. It is caused by TLS headers and padding added between TCP and application layer.

Packet Size							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Packet Size-NoTLS(KBytes)	4.91	2.54	3.21	5.69	3.72	4.83	2.47
Packet Size-TLS(KBytes)	7.20	4.80	5.47	7.97	6.00	7.11	4.77
Total TLS Packet Overhead(%)	47%	89%	70%	40%	61%	47%	93%
Registration-only Packets Overhead(%)	4%	7%	5%	3%	5%	4%	8%

Table 5.18: The traffic overhead of TLS with ECDHE-AES256-GCM-SHA384 cipher suite for overall registration process and the registration-only packets.

Table 5.19 denotes the number of messages exchanged. There are around a 19% to 28% increase in message count for the total registration process across the network functions.

On the contrary, TLS sends fewer messages in the registration-only operation that is mainly concerned with the WINDOWS UPDATE and SETTINGS messages of HTTP2 protocol. We observed that it decreased from 10% to 16% for different NFs as shown in table 5.20.

Time overhead in table 5.21 shows how long it took each NF to register the profile

Total Number of Message							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Messages-NoTLS(Count)	18.48	16.68	18.56	18.56	18.48	18.76	16.68
Messages-TLS(Count)	22.42	20.81	22.11	22.15	22.42	22.42	21.28
TLS Message Overhead(%)	21%	25%	19%	19%	21%	20%	28%

Table 5.19: The message count overhead added to the overall registration process by TLS-ECDHE-AES256-GCM-SHA384 cipher suite.

Number of Registration-only Message							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Messages-NoTLS(Count)	15.48	13.68	15.56	15.56	15.48	15.76	13.68
Messages-TLS(Count)	13.41	11.80	13.10	13.14	13.41	13.41	12.26
Registration-only Message Overhead(%)	-13%	-14%	-16%	-16%	-13%	-15%	-10%

Table 5.20: The decrease in total number of messages in registration-only subprocess with ECDHE-AES256-GCM-SHA384 cipher suite.

with the NRF when the Free5GC slice is active and NRF accepts requests from the slice components. TLS cipher suite of ECDHE-AES256-GCM-SHA384 adds around 60% to 70% overhead for intra-slice communication.

Total Registration Time							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Time-NoTLS(ms)	14.89	16.05	20.96	13.16	23.58	21.08	26.05
Time-TLS(ms)	25.05	27.24	36.25	22.63	39.74	33.58	36.59
Total Time Overhead(%)	68%	70%	73%	72%	69%	59%	40%

Table 5.21: The time overhead TLS-ECDHE AES256-GCM-SHA384 on the whole registration process.

5.2.3 AES128-GCM-SHA256

This experiment relates to evaluating the performance of the TLS cipher suite of AES128-GCM-SHA384 with the ECDHE key exchange algorithm. The main difference between AES128 with AES256 is the key length used for encryption that decryption procedures. The numbers 128 and 256 denote the key in bit length for the two cipher suites.

Table 5.22 shows the packet size values in KiloBytes for the whole registration process for both with no TLS and with TLS ECDHE-AES128-GCM-SHA256 cipher suite. The total packet size overhead is mainly because of TLS handshake traffic. TLS record protocol adds from 3% to 6% packet size overhead to registration-only sub-process.

Packet Size							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Packet Size-NoTLS(KBytes)	4.96	2.57	3.21	5.70	3.80	4.89	2.51
Packet Size-TLS(KBytes)	7.23	4.82	5.47	7.97	6.03	7.13	4.75
Total TLS Packet Over-head(%)	46%	88%	70%	40%	59%	46%	89%
Registration-only Packets Overhead(%)	3%	6%	5%	3%	3%	3%	6%

Table 5.22: The packet size overhead of TLS with ECDHE-AES128-GCM-SHA256 cipher suite for the total registration process and the registration-only packets.

Regarding the number of messages transmitted in the whole registration procedure, the results show that the overhead ranges from 16% to 23% across different NFs as shown in Table 5.23. Notably, there are fewer messages in the registration-only sub-process as shown in Table 5.24.

Table 5.25 illustrates the nominal values in milliseconds for total registration time for both TLS and no TLS implementation. The TLS-ECDHE-AES128-GCM-SHA256

Total Number of Message							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Messages-NoTLS(Count)	19.24	17.16	18.60	18.64	18.92	19.60	17.32
Messages-TLS(Count)	22.87	21.08	22.22	22.22	22.51	22.74	21.05
TLS Message Overhead(%)	19%	23%	19%	19%	19%	16%	22%

Table 5.23: The sent messages overhead due to using TLS-ECDHE-AES128-GCM-SHA256 cipher suite for the whole registration operation.

Number of Registration-only Message							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Messages-NoTLS(Count)	16.24	14.16	15.60	15.64	15.92	16.60	14.32
Messages-TLS(Count)	13.85	12.07	13.20	13.21	13.45	13.73	12.03
Registration-only Message Overhead(%)	-15%	-15%	-15%	-16%	-16%	-17%	-16%

Table 5.24: The sent messages overhead due to using TLS-ECDHE-AES128-GCM-SHA256 cipher suite for the registration-only operation.

increases 30% to 50% time overhead to the different slice components.

Total Registration Time							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Time-NoTLS(ms)	33.78	25.44	27.44	24.50	26.21	28.82	29.05
Time-TLS(ms)	47.03	35.27	36.67	38.82	37.85	41.24	38.64
Total Time Overhead(%)	39%	39%	34%	58%	44%	43%	33%

Table 5.25: The time overhead TLS-ECDHE AES128-GCM-SHA256 adds on the whole registration process.

5.3 Results Comparison

5.3.1 RSA

This section illustrates and compares the testing results between different TLS1.2 cipher suites with each key exchange mechanism RSA and ECDHE. Results show error bars that corresponds to 95% confidence interval around the mean on the different network performance parameters including time, packet size, and message count.

The total packet size added by TLS1.2 in figure 5.1 shows that the cipher suites of AES128-GCM-SHA265, and AES256-GCM-SHA384 have added the same amount of the packet size overhead to all network functions. Nonetheless, the 3DES-EDE-CBC-SHA cipher suite added more packet size overhead for the majority of the network functions. There is a significant extra overhead on AUSF, NSSF, SMF, and UDR.

It is good to note that the three cipher suites employ the same key exchange scheme, RSA, to exchange the symmetric key and make the TLS handshake. Therefore, the reason why 3DES imposed extra traffic overhead is because of the TLS record protocol mode of operations. To clarify this we examined the registration only packet overhead, where we consider the TLS overhead after the TLS handshake is completed and the application data are encrypted and decrypted with TLS. It will be elaborated in the remaining part of this section.

Figure 5.2 shows that 3DES add significantly more packet size overhead than AES128 and AES256 in registration-only messages for the majority of the network functions. The difference stems from the mode of the operation of the employed cipher suites. 3DES uses CBC, which splits the plain text into blocks and encrypts them with a key and XOR it with cipher block of the previous plain text. Plain text block number three encryption depends on and is fed with the ciphertext of a plain text block number two. However, the GCM mode operates in parallel and carries out

Figure 5.1: The comparison of total packet size overhead on the registration process with TLS cipher suites and RSA key exchange.

the cryptography operation for each block independently so it adds less encryption overhead than CBC mode that encryption and decryption operate sequentially leading to add extra overhead at TLS layer compared to GCM mode.

Figure 5.3 compares the percentage of extra messages the three cipher suites bring to the whole NF registration process. We can conclude that all three cipher suites using the RSA key exchange add around 15% to 25% more messages on average across all the network functions. The increase is realized mainly due to the extra transactions needed to perform the TLS handshake protocol.

As figure 5.4 depicts, there are fewer messages communicated between different NFs and NRF for the registration-only subprocess for all three cipher suites, and there is no significant difference among all the cipher suites. Notably, the decrease is mainly influenced by the HTTP2 mechanism that sends fewer HTTP2 packets such as HTTP2 SETTINGS and WINDOW UPDATE.

Figure 5.2: The comparison of the packet size overhead for registration-only sub-process with TLS cipher suites and RSA key exchange.

Figure 5.3: The comparison of the extra message overhead on the total registration process with TLS cipher suites and RSA key exchange.

Figure 5.5 shows the comparison of the total registration time overhead of TLS for three different cipher suites of 3DES-EDE-CBC-SHA, AES256-GCM-SHA384,

Figure 5.4: The comparison of decrease in number of messages for the registration-only sub-process with TLS cipher suites and RSA key exchange.

and AES128-GCM-SHA256. Some network functions using AES128 take less time than AES256 to undergo the registration process and get the profile subscribed in the NRF. The figure shows there is a significant difference in terms of time duration for some of the NFs.

AES128-GCM-SHA256 carries out encryption and decryption using a master symmetric key of 128 bits in length whereas AES256-GCM-SHA384 performs the same tasks with a key length of 256. Although using a longer key provides stronger security strength, it consumes more time to implement cryptography operations.

It is noteworthy that 3DES-EDE-CBC-SHA uses a key length of 168 but with a security strength equivalence of 112 key bits long. Although 3DES uses almost the same key size as AES128, it adds significantly more latency overhead to the total registration process as compared to AES128 in some cases.

The reason is that the 3DES deploys CBC mode that has no parallelism in ciphering and deciphering plain text block which is the main reason it is quite slow comparatively with GCM (Galois Counter Mode) operation. In GCM, there is a random number called nonce encrypted to produce either 128 bits or 256 bits key length. Then the keys are XORed to the plain text block to produce ciphertext blocks. It is notable that despite the CBC mode, in GCM the encryption of each block of plain text does not depend on the previous ciphertext. Thus, there is parallelism as well as optimization in message encryption(encrypting the random number instead of the pain text) mechanism in GCM that make its cryptography operation faster than CBC.

Figure 5.5: The comparison of the time overhead by the TLS cipher suites on the total registration process.

5.3.2 ECDHE

This section compares performance overhead the same cipher suites, including AES128-GCM-SHA265, AES256-GCM-SHA384, and 3DES-EDE-CBC-SHA, but

with ECDHE key exchange mechanism bring to the NF registration with NRF procedure.

Figure 5.6 compares the total packet size added by the three cipher suites to the registration process. The measurements show that there is no significant difference for added packer overhead for the cipher suites of AES128-GCM-SHA265, AES256-GCM-SHA384; however, the 3DES-EDE-CBC-SHA adds slightly more than the other two cipher suites in some cases. The reason is related to the different mode of operations the AES and 3DES use in the TLS record protocol.

Figure 5.6: The comparison of total packet size overhead on the registration process with TLS cipher suites and ECDHE key exchange.

Figure 5.7 shows that 3DES adds significantly more packet size overhead than AES128 and AES256 in registration-only messages for the majority of the network functions. The difference stems from the mode of the operation 3DES and AES cipher suites employ while carrying out the cryptography operations.

Figure 5.7: The comparison of the packet size overhead for registration-only sub-process with TLS cipher suites and ECDHE key exchange.

Figure 5.8 shows the percentage of extra messages the three cipher suites bring to the total NF registration process compared to the noTLS case. There are no significant difference for the majority of the NFs. The increase is realized solely due to the extra transactions needed to perform the TLS handshake protocol.

However, in the TLS scenario in either of the three cipher suites, there are fewer messages exchanged between different NFs and NRF for the registration-only subprocess. The reason is that because the HTTP2 mechanism sends fewer HTTP2 packets such as HTTP2 SETTINGS and WINDOW UPDATE. There is no significant difference in decrease between the three cipher suites for the majority of NFs as shown in the figure 5.9.

Figure 5.10 compares the TLS time overhead for the cipher suites. Almost all the network functions using AES128 take less time than AES256 to undergo the registration process and get the profile subscribed in the NRF. The graph shows there is a significant difference in terms of time duration for the majority of the

Figure 5.8: The comparison of the extra message overhead on the total registration process with TLS cipher suites and ECDHE key exchange.

Figure 5.9: The comparison of decrease in number of messages for the registration-only sub-process with TLS cipher suites and ECDHE key exchange.

NFs. The reason why AES256 has more time overhead than AES128 is that because AES256 performs the cryptography operations such as encryption, decryption, and

hash function with 256 bits key long whereas AES128 uses a 128 bits long key to carry out the same operation. The longer key has more processing time but provides more security strength.

On a different note, 3DES-EDE-CBC-SHA has the same time overhead compared to AES256-GCM-SHA384, although it uses a shorter key, 168 bits long, length than AES256. As elaborated in the earlier section related to the RSA key exchange, there is parallelism as well as optimization in message encryption(encrypting the random number instead of the pain text) mechanism in GCM that make its cryptography operation faster than CBC.

Figure 5.10: The comparison of the time overhead by the TLS cipher suites with ECDHE key exchange on the total registration process.

Comparing RSA with ECDHE, although we have noticed slight differences of overhead in metrics such as total packet size, registration-only packet size, and total registration time in some NFs, overall the three cipher suites using the different key exchange methods add similar overhead to the NF registration process.

5.4 Test with TLS 1.3 RSA-AES128-GCM-SHA256

This section aims to evaluate the impact of TLS 1.3 performance on network function registration with the network repository function procedure. We selected RSA-AES128-GCM-SHA256 cipher suite because it is one of the commonly used cipher suite for both cases of TLS1.2 and TLS1.3.

Pursuing the same methodology applied for TLS1.2, we let each network function register their profile with NRF twenty-five times for both cases of when the slice components exchange the control messages (CP) on TCP and when they exchange CP messages on TLS1.3 with RSA-AES128-GCM-SHA256 cipher suite. It aims to provide us a baseline, that is noTLS scenario, to analyze and compare our measurements with the TLS1.3 scenario. Notably, we allowed the registration for one NF at a time to isolate the measurement process. The following tables illustrate how much TLS 1.3 adds and imposes overhead in terms of time and latency to the total registration process, packet size, and the number of messages exchanged between NF and NRF for the registration process.

Packet Size							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Packet Size-NoTLS(KBytes)	4.89	2.54	3.17	5.68	3.70	4.81	2.48
Packet Size-TLS(KBytes)	7.09	4.71	5.35	7.89	5.89	7.00	4.65
Total TLS Packet Overhead(%)	45%	85%	68%	39%	59%	46%	87%
Registration-Only Packet Overhead(%)	1.9%	2.3%	2%	1.8%	2.2%	1.7%	2.5%

Table 5.26: Total packet size and registration-only size overhead by the TLS1.3-RSA-AES128-GCM-SHA256 cipher suite.

Total Number of Message							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Messages-NoTLS(Count)	18.12	16.72	18.04	18.32	18.24	18.36	16.88
Messages-TLS(Count)	20.98	19.56	20.61	21.20	21.08	21.09	19.71
TLS Message Overhead(%)	16%	17%	14%	16%	16%	15%	17%

Table 5.27: The sent messages overhead due to using TLS1.3-RSA-AES128-GCM-SHA256 cipher suite for whole registration process.

Number of Registration-only Message							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Messages-NoTLS(Count)	15.12	13.72	15.04	15.32	15.24	15.36	13.88
Messages-TLS(Count)	12.98	11.55	12.60	13.20	13.07	13.07	11.77
Registration-only Message Overhead(%)	-14%	-16%	-16%	-14%	-14%	-15%	-15%

Table 5.28: The sent messages overhead due to using TLS1.3-RSA-AES128-GCM-SHA256 cipher suite for the registration-only.

Total Registration Time							
NF Name	AMF	AUSF	NSSF	PCF	SMF	UDM	UDR
Time-NoTLS(ms)	22.59	19.18	19.90	15.95	18.20	22.28	19.94
Time-TLS(ms)	31.81	26.24	24.65	23.52	26.46	28.51	27.36
Total Time Overhead(%)	41%	37%	24%	47%	45%	28%	37%

Table 5.29: The time overhead TLS1.3-RSA-AES128-GCM-SHA256 adds to the overall registration process.

The following figures show and compare the measurement results when the network functions to register the profile with NRF with TLS1.2 and TLS1.3. We already tested TLS1.2 with the same cipher suite, RSA-AES128-GCM-SHA256, in the earlier section of this chapter.

Figure 5.11 shows that both TLS 1.3 and TLS 1.2 exchanged extra messages compared to when the NF registration is performed with no TLS. It mainly stems

from the TLS handshake protocol.

Notably, TLS1.3 includes fewer extra messages, which ranges from 14% to 17%, than TLS1.2, which ranges from 19% to 25%, across different network functions. The reason is that because TLS1.3 completes the handshake protocol in one round trip whereas TLS1.2 carries out the TLS handshake in 2 round trips.

Figure 5.11: The comparison of the extra messages TLS1.2 and TLS1.3 have on the NF registration process compared to when the same procedure occurs over TCP, no TLS case.

There are fewer messages exchanged for registration only sub-process both TLS 1.2 and TLS 1.3 compared to no TLS scenario as shown in figure 5.12. This is mainly as a result of the HTTP2 mechanism that sends fewer HTTP2 packets such as HTTP2 SETTINGS and WINDOW UPDATE. The measurements show that there are no significant difference for this metric between TLS12 and TLS13 for the same cipher suite of AES128-GCM-SHA256.

Figure 5.13 compares the total packet size added by TLS1.2 and TLS1.3 with the same cipher suite of AES128-GCM-SHA256 to the registration process. Although

Figure 5.12: The comparison of decrease in number of the messages by TLS12 and TLS13 for the registration-only messages.

there are fewer messages exchanged in TLS13 compared to TLS12, the same amount of the information is exchanged in both TLS versions. Therefore there is no significant difference in terms of total packet size overhead added between them.

Figure 5.13: The comparison of total packet size overhead between TLS12 and TLS13.

Regarding registration only sub-process, the TLS1.3 adds significantly less packet overhead than TLS1.2 that mainly stems from the HTTP2 mechanism as shown in the graph 5.14.

Figure 5.14: The comparison of the total packet size overhead between TLS 12 and TLS13.

Figure 5.15: The comparison of the total time overhead the TLS12 and TLS13 add to the registration process.

Figure 5.15 compares the total latency the TLS12 and TLS13 adds to the registration procedure. Although the measurements have shown that TLS13 has significantly less time overhead on SMF and UDR, there is no significant difference in terms of time overhead between the TLS version for the rest of the NFs.

Chapter 6

TLS Impact On The UE Initial Registration With a 5G Core Slice

This chapter focuses on finding out the effect distinct TLS cipher suites can have on the duration of when UE is switched on and wants to register itself to a 5G core slice. With 5G technology ultra-Low-latency capacity, it is expected to facilitate an extra boost to a wider spectrum of related use cases and applications that are sensitive to the latency like autonomous vehicles, e-health, and the SCADA industry [55] . At the same time, 3GPP [14] recommends security measures for users' information while they travel across the default network slice for different purpose of control plane signaling. UE registration occurs when a UE is switched on and wants to gain access to the network. As mentioned earlier, some UEs can tolerate very little latency when they want to connect to a 5G network. Having security measures implemented between the network functions adds extra processing time. This chapter conducts multiple experiments each using TLS with different cipher suites that aims to demonstrate an example of the impact of TLS adoption.

6.1 Commissioning 5G Core, UE, gNodeB and Verifying UE Provisioning in UDM

Initially, we need to find out how long it takes a UE to make the initial registration to a 5G slice in the context where there is no TLS for authentication and encryption of the control plane traffic. Since we use docker-compose we need to run all network function images after they have been created with the "docker-compose build" command. After we verified that all the network functions have registered their profile to the network repository function any user can access the slice through N1 and N2 interfaces. Those interfaces carry their traffic over SCTP and NGAP protocols and are terminated at Access and Mobility Function (AMF). User equipment information need to be provided to the Unified Data Management (UDM). The UDM is a significant node that stores all subscribers' data and profiles. In the experiments, we registered the UE's information to the UDM via the free5GC web console.

6.2 UE Initial Registration with a 5G SBA Core Slice

We deploy the gnbssim that emulates a UE and gNodeB together. By the standards, although there is a logical interface between a UE and a 5G core slice that is called N2, it has to reach the slice through gNodeB so there has to be an interface between gNodeB and AMF. It is called the N1 interface. AMF as one of the default network functions listens to the N1 interface and deals with control plane traffics triggered from both UEs and gNodeB. Since we use two separate open-source projects namely Free5GC and gnbssim to emulate a 5G core slice and UE/gNodeB respectively, we configured the gnbssim related source file, called example.json that includes routing

parameters. These parameters are GTPuAddr and GTPuIfname. "GTPuAddr" allows gNodeB route and exchange control plane traffic via N1 interface towards Free5GC active slice. Beside the interface and since we run the experiment on the VM, we shall ensure we assigned the virtual interface to the GTPuIfname parameter.

The testbed can grant the UE access to the network when the UE is switched on and make the initial interaction to the operative Free5GC slice. We examined the related collection in the MongoDB to verify the UE is successfully registered itself with the Free5GC active slice.

We examined the packets being transmitted by all the virtual interfaces associated with the NFs and gnbSim.

The registration duration accounts for the time the UE initiates registration request through AMF until the time UE receives the registration acceptance through AMF. Thus, to calculate the registration duration we take the timestamp of the initial control message and subtract it from the time the registration is completed. The capturing points are illustrated in figure 6.1.

Figure 6.1: Simplified Representation of a UE Registration to a 5G Core Slice and specifying the capturing points to calculate the UE initial registration time.

6.3 Test with RSA key Exchange

The remaining sections of this chapter focus on performing UE initial registration with different TLS1.2 cipher suites including 3DES-EDE-CBC-SHA, AES128-GCM-SHA256, and AES256-GCM-SHA384. It aims to figure out how much time overhead each TLS cipher suite adds to UE initial registration. We developed a bash script that automated the UE registration process with an active Free5gc slice. It aimed to reduce human intervention and error as much as possible. We measured the registration time when each of three cipher suites is active in the slice.

6.3.1 3DES-EDE-CBC-SHA

This section runs tests of the UE initial registration with a slice when NFs encrypts all control plane messages using the cipher suite of 3DES-EDE-CBC-SHA before they dispatch them to another network functions within an operative slice.

Since the docker-compose tool was used to build the 5G core slice, we build all the NFs images. We initiated the UE initial registration from gnbssim and verified that it is successfully done.

Table 6.1 shows the 3DES-EDE-CBC-SHA added 20.8% time overhead to the UE initial registration to the active slice.

UE Initial Registration (Seconds)	
No TLS	0.7134
TLS-3DES	0.862
Time Overhead	20.8%

Table 6.1: UE registration process for both scenarios of no TLS and TLS with the cipher suite of 3DES-EDE-CBC-SHA.

6.3.2 AES256-GCM-SHA384

We tested the UE initial registration when our emulated 5G slice is active and perform cryptography operations on the control plane traffic with the cipher suite of AES256-GCM-SHA384. We followed the same preparation and testing sequences we carried out in the previous cipher suite.

The table 6.2 shows the AES256-GCM-SHA384 added 41.95% time overhead to the UE initial registration to the active slice.

UE Initial Registration (Seconds)	
No TLS	0.7134
TLS-AES256	1.013
Time Overhead	41.95%

Table 6.2: UE registration process for both scenarios of no TLS and TLS with the cipher suite of AES256-GCM-SHA384.

6.3.3 AES128-GCM-SHA256

We tested the UE registration with the slice that its NFs encryp the traffic using the AES128-GCM-SHA256 cipher suite.

Table 6.3 shows the AES128-GCM-SHA256 added 20.4% time overhead to the UE initial registration to the active slice.

UE Initial Registration (Seconds)	
No TLS	0.7134
TLS-AES256	0.859
Time Overhead	20.41%

Table 6.3: UE registration process for both scenarios of no TLS and TLS with the cipher suite of AES128-GCM-SHA256.

To sum up, Figure 6.2 shows the results all three suites explained above along the case when no TLS is applied for the comparison purpose. This figure shows that the AES128-GCM-SHA256 suite gives the least UE registration time. AES256-GCM-SHA384 causes UE to spend 41.95% more time relative to the case when the slice core components do not use TLS. In the second scenario where 3DES-CBS-SHA is used, the time overhead added to the initial registration flow increased by 20.8% relative to the case of no TLS in place. AES128-GCM-SHA256 imposed the same impact, 20.41%, as the 3DES one although it uses a key with more security strength. As expected AES128-GCM-SHA256 added less time overhead compared to AES256-GCM-SHA384 because it perform the cryptography operations with key with a shorter length.

Figure 6.2: Time overhead of TLS cipher suites of 3DES-EDE-CBC-SHA, AES128-GCM-SHA256, and AES256-GCM-SHA384 on UE initial registration. UE registration with no TLS is considered as the baseline.

Chapter 7

TLS Impact On the UE PDU Session Establishment With a 5G Core Slice

In the 5G technology, there is a control plane operation that facilitates end-to-end data traffic (user plane) connectivity between the user equipment and particular data service and network. The control plane procedure is called PDU(Packet Data Unit) session establishment. It creates a logical connection through the UE, gNodeB, and UPF called GTP tunnel that is used to allow the flow of the user plane traffic from the UE to a specific data network (DN) and vice versa. Nonetheless, UE has to initiate and go through the PDU session establishment procedure before reaching its subscribed data traffic that involves a 5G core slice that includes the control plane network functions. UE triggers the PDU session request for several cases such as, setting up a new PDU session, roaming from PDN in 4G/LTE to PDU session in 5G, handing over a PDU session between a 3GPP and non 3GPP access network and, in emergency scenario. This chapter focuses on the scenario where a UE sends a new PDU session establishment request towards a 5G active slice. We could not find studies on measuring the impact of TLS performance on 5G control plane operations that are triggered by UE such as PDU session establishments as it is considered one

of important KPIs in 3GPP [71].

7.1 Commissioning 5G Core, UE, gNodeB and Verifying UE Provisioning in UDM

We build and commission a testbed consisting of different sections including UE, gNodeB, 5G core slice, and service to study the impact of different cipher suites operations of TLS1.2 on the PDU session establishment flow. We used gnbssim to simulate UE and gNodeB. It is vital to ensure the gnbssim is configured with correct IP addressing so that it communicates with the 5G core slice that is emulated with Free5GC open-source project. The UE exchange signalling messages with AMF of the core via gNodeB using the N1 interface. Also, gNodeB talks to AMF via N2 interface that exchanges control plane signaling messages related to scenarios such as UE context management, PDU session/resource management procedures. SCTP(NGAP) (Stream Control Transport Protocol) protocol is used between RAN and a 5G core slice. It functions at the transport layer and handles multiple concurrent streams whereas TCP handles only a stream per connection. After finishing the configuration of the gnbssim we instantiated 5G core slice using docker-compose.

The second step of the testbed preparation is to register a UE with the Free5GC active slice. It is noteworthy that a UE can not tackle a PDU session to connect to any of its subscribed services unless it is registered and granted access to the 5G network slice. To let a UE gain access to the slice we should have initially provisioned its subscription and profile with UDM. Once we completed the UE provisioning in UDM using Free5GC WebGUI access, we let the UE make the initial registration with the slice.

Figure 7.1 depicts the control plane (CP) signaling and the user plane (UP) traffic flows occurring between distinct 5G network including CP NFs, UP NFs, and UE/gNodeB. The dashed lines denote CP flows and the solid lines denote UP traffic.

7.3 Test with RSA key Exchange

After setting up the testbed and executing several PDU session establishment requests we come to the point to measure the time it takes a UE to receive its user plane traffic after sending a service request that is encapsulated in SCTP(NGAP) message and routed to AMF via gNodeB.

Figure 7.2 indicates the points in the process, circled in blue, where we consider the start time and the end time when is the gNodeB sends the acceptance message to AMF and the initial user plane traffic flows between UE and the service that is connected to UPF by N6 interface.

In the experiments, we calculated the delta time for the following scenarios:

- Intra-slice communication using no TLS
- Intra-slice communication using 3DES-EDE-CBC-SHA
- Intra-slice communication using AES128-GCM-SHA256
- Intra-slice communication using AES256-GCM-SHA384

We conducted the PDU session establishment to the active slice with no TLS and measure the PDU establishment time as the baseline of our calculation. We conduct the same experiment when the NFs of the slice secure their communication with each of the aforementioned cipher suites and measure the PDU establishment time to calculate the time overhead for each cipher suite.

Figure 7.2: Simplified representation of the flows related to establishing a PDU session and GTP-U. It illustrates the start and end capturing points to calculate the PDU session Establishment time in TLS and no TLS scenarios.

7.3.1 3DES-EDE-CBC-SHA

This section aims to figure out how much time overhead the 3DES-EDE-CBC-SHA cipher suite has on the PDU session flow. We instantiated the slice using the docker-compose tool and we captured the messages with tcpdump while the slice being composed to reassure that all the NFs registered the profile with NRF on the 3DES-EDE-CBC-SHA cipher suite. At this step, we confirmed the slice is operative. We ran a light-weight python HTTP server to simulate a compute node. It exists on the same virtual machine as the Free5GC; however, we assigned IP addressing to the python server to be reachable either of UPF1 and UPF2 via the N6 interface.

Table 7.1 shows the 3DES-EDE-CBC-SHA added 27.5% time overhead to the UE PDU session establishment to the active slice.

PDU Session Establishment Procedure Duration (Seconds)	
No TLS	0.299
TLS-AES256	0.381
Time Overhead	27.5%

Table 7.1: PDU session establishment process for both scenarios of no TLS and TLS with the cipher suite of 3DES-EDE-CBC-SHA.

7.3.2 AES256-GCM-SHA384

This section aims to measure and reflect the impact of the AES256-GCM-SHA384 on the control plane operation of PDU session that leads a UE to connect to the data network and consume its subscribed service. We followed the same preparation and testign sequences as the previous cipher suite.

Table 7.2 includes both nominal values for no TLS and with TLS cipher suite of AES256-GCM-SHA384. It shows the AES256-GCM-SHA384 added 42.5% time overhead to the UE PDU session establishment to the active slice.

PDU Session Establishment Procedure Duration (Seconds)	
No TLS	0.299
TLS-AES256	0.425
Time Overhead	42.5%

Table 7.2: PDU session establishment process for both scenarios of no TLS and TLS with the cipher suite of AES256-GCM-SHA384.

7.3.3 AES128-GCM-SHA256

At last, we followed the same sequences performed for the last two cipher suites and commissioned the Free5GC with TLS AES128-GCM-SHA256.

Table 7.3 shows the AES128-GCM-SHA256 added 19.5% time overhead to the

UE PDU session establishment to the active slice.

PDU Session Establishment Procedure Duration (Seconds)	
No TLS	0.299
TLS-AES256	0.357
Time Overhead	19.5%

Table 7.3: PDU session establishment process for both scenarios of no TLS and TLS with the cipher suite of AES128-GCM-SHA256

To sum up, Figure 7.3 shows the results all three suites explained above along the case when no TLS is applied for the comparison purpose. This figure shows that the AES128-GCM-SHA256 suite gives the least UE PDU session establishment time. AES256-GCM-SHA384 increases the PDU procedure by 42.5% compared to AES128-GCM-SHA256 that is 19.5%. AES256 performs cryptography operations in longer while because it encrypts and decrypts messages with a key length of 256 bits whereas AES128 uses a shorter key length, 128 bits. 3DES-EDE-CBC-SHA adds 27.5% overhead that is somehow close to AES128 though they use different cryptography mode of block operations that are CBC(Cipher-Block Chaining) and GCM(Galois/Counter Mode). AES256 provides stronger security strength because of its longer key length compared to AES128 and 3DES.

Figure 7.3: Time overhead of TLS cipher suites of 3DES-EDE-CBC-SHA, AES128-GCM-SHA256, and AES256-GCM-SHA384 on UE PDU session establishment. UE PDU session establishment with no TLS is considered as the baseline.

Chapter 8

Conclusions

8.1 Summary

This research aimed to study the impact of TLS operations, including cryptography functions and TLS handshake for authentication, on control plane procedures within different stages of a 5G slice lifecycle. The 5G core procedures included network function (NF) registrations with network repository function (NRF), initial registration of user equipment (UE) to gain access to a 5G slice, and PDU session establishment of a UE to access services. We found out how much impact TLS1.2 with different cipher suites including 3DES-EDE-CBC-SHA, AES128-GCM-SHA256, and AES256-GCM-384, have on different key performance parameters of time, packet size, and message count in inter-communication of 5G core slice. We used two different key exchange mechanisms of RSA and ECDHE for each TLS experiment. Furthermore, we extended our experiment to TLS1.3 with the RSA-AES128-GCM-SHA256 cipher suite.

We took an empirical approach in the research to figure out how much TLS affects different control plane operations of a 5G core slice. As expected cipher

suite of AES256-GCM-SHA384 adds more time overhead compared to AES128-GCM-SHA256 which mainly stems from the longer key AES256 uses to make the cryptography operations.

We used 3DES-EDE-CBC-SHA for comparison purposes only. It has been shown that 3DES has deficiencies in terms of security and performance¹.

We note that the results of our experiments are specific to the environment used and the crypto library of Golang used for this study.

Results showed that TLS1.3 adds slightly less time overhead and number of messages than TLS1.2 using AES128-GCM-SHA256 cipher suite for NF registration process. It is mainly caused by TLS1.3 handshake protocol that processes in one round trip whereas TLS1.2 performs it in two-round trips. Also, we realized that TLS1.3 adds less packet size overhead than TLS1.2 to the registration-only messages. For both TLS versions, there are less number of messages sent in registration-only subprocess. This research demonstrated the TLS performance impact on important network KPIs related to different 5G control plane operations. It provides insight to the research communities about the impact of securing intra-slice communication with TLS. Furthermore, the research findings complement the TLS related works such as TLS impact on SIP, and TLS analysis in ToR.

¹<https://csrc.nist.gov/News/2017/Update-to-Current-Use-and-Deprecation-of-TDEA>

Bibliography

- [1] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The next generation wireless access technology*. Academic Press, 2020.
- [2] N. Alliance, “5G white paper,” *Next generation mobile networks, white paper*, vol. 1, 2015.
- [3] 3GPP, “General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access,,” TS 23.401 V16.0.8, 3GPP, December 2014.
- [4] P. Porambage, Y. Miche, A. Kalliola, M. Liyanage, and M. Ylianttila, “Secure keying scheme for network slicing in 5g architecture,” in *2019 IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 1–6, IEEE, 2019.
- [5] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, “5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges,” *Computer Networks*, vol. 167, p. 106984, 2020.
- [6] 3GPP, “Technical Specification Group Services and System Aspects; Study on Architecture for Next Generation System;Stage 2;(R14),” TS 23.799 V14.0.0, 3GPP, December 2016.

- [7] P. Rost, A. Banchs, I. Berberana, M. Breitbach, M. Doll, H. Droste, C. Manweiler, M. A. Puente, K. Samdanis, and B. Sayadi, “Mobile network architecture evolution toward 5G,” *IEEE Communications Magazine*, vol. 54, no. 5, pp. 84–91, 2016.
- [8] 3GPP, “Management and orchestration; Provisioning; (R16),” TS 28.531 V16.5.0, 3GPP, March 2020.
- [9] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo, and R. Jain, “Network slicing for 5G: Challenges and opportunities,” *IEEE Internet Computing*, vol. 21, no. 5, pp. 20–27, 2017.
- [10] M. Villari, W. Zimmermann, and K.-K. Lau, *Service-Oriented and Cloud Computing*. Springer, 2014.
- [11] V. Yazıcı, U. C. Kozat, and M. O. Sunay, “A new control plane for 5G network architecture with a case study on unified handoff, mobility, and routing management,” *IEEE Communications Magazine*, vol. 52, no. 11, pp. 76–85, 2014.
- [12] T. Dierks and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2,” RFC 5246, RFC Editor, August 2008. <http://www.rfc-editor.org/rfc/rfc5246.txt>.
- [13] 3GPP, “Technical Specification Group Services and System Aspects and Procedures for the 5G System (5GS); Stage 2; (R16),” TS 23.502 V16.6.0, 3GPP, September 2020.
- [14] 3GPP, “Network Domain Security and Authentication Framework (R16),” TS 33.310 V16.5.0, 3GPP, September 2020.

- [15] 3GPP, “Security architecture and procedures for 5G system (R16),” TS 33.501 V16.2.0, 3GPP, March 2020.
- [16] 3GPP, “System architecture for the 5G System (5GS); Stage 2; (R16),” TS 23.501 V16.5.1, 3GPP, August 2020.
- [17] H. Yu, H. Lee, and H. Jeon, “What is 5G? Emerging 5G mobile services and network requirements,” *Sustainability*, vol. 9, no. 10, p. 1848, 2017.
- [18] I. Vision, “Framework and overall objectives of the future development of int for 2020 and beyond,” *International Telecommunication Union (ITU), Document, Radiocommunication Study Groups*, 2015.
- [19] R. Trivisonno, X. An, and Q. Wei, “Network slicing for 5G systems: A review from an architecture and standardization perspective,” in *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 36–41, IEEE, 2017.
- [20] Alliance, NGMN, “Description of network slicing concept,” *NGMN 5G P*, vol. 1, no. 1, 2016.
- [21] E. Kapassa, M. Touloupou, A. Mavrogiorgou, A. Kiourtis, D. Giannouli, K. Katsigianni, and D. Kyriazis, “An Innovative eHealth System Powered By 5G Network Slicing,” in *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, pp. 7–12, IEEE, 2019.
- [22] T. Taleb, P. A. Frangoudis, I. Benkacem, and A. Ksentini, “CDN slicing over a multi-domain edge cloud,” *IEEE Transactions on Mobile Computing*, 2019.

- [23] E. NFV-EVE, “Network Functions Virtualisation (NFV) Release 3; Evolution and Ecosystem; Report on Network Slicing Support with ETSI NFV Architecture Framework,” 2017.
- [24] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, “Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges,” *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, 2017.
- [25] Y.-i. Choi and N. Park, “Slice architecture for 5g core network,” in *2017 Ninth international conference on ubiquitous and future networks (ICUFN)*, pp. 571–575, IEEE, 2017.
- [26] J. Kim, D. Kim, and S. Choi, “3GPP SA2 architecture and functions for 5G mobile communication system,” *ICT Express*, vol. 3, no. 1, pp. 1–8, 2017.
- [27] A. Freier, P. Karlton, and P. Kocher, “The ssl protocol version 3.0. draft-ietf-tls-ssl-version3-00. txt,” 1996.
- [28] T. Dierks and C. Allen, “The TLS Protocol Version 1.0,” RFC 2246, RFC Editor, January 1999. <http://www.rfc-editor.org/rfc/rfc2246.txt>.
- [29] S. Thomas, “SSL and TLS essentials,” *New Yourk*, vol. 3, 2000.
- [30] W. Stallings, L. Brown, M. D. Bauer, and A. K. Bhattacharjee, *Computer security: principles and practice*. Pearson Education Upper Saddle River, NJ, USA, 2012.
- [31] “Wikimedia foundation: Transport layer security,” Dec 2020.
- [32] Microsoft, “Transport layer security protocol - win32 apps,” 2010.

- [33] P. Morrissey, N. P. Smart, and B. Warinschi, “The TLS handshake protocol: A modular analysis,” *Journal of Cryptology*, vol. 23, no. 2, pp. 187–223, 2010.
- [34] A. Annunziato, “5G vision: NGMN-5G initiative,” in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pp. 1–5, IEEE, 2015.
- [35] M. Series, “IMT Vision—Framework and overall objectives of the future development of IMT for 2020 and beyond,” *Recommendation ITU*, vol. 2083, 2015.
- [36] C. Shen, E. Nahum, H. Schulzrinne, and C. P. Wright, “The impact of TLS on SIP server performance: Measurement and modeling,” *IEEE/ACM Transactions on Networking*, vol. 20, no. 4, pp. 1217–1230, 2012.
- [37] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, *et al.*, “Sip: session initiation protocol,” 2002.
- [38] E. Coulibaly and L. H. Liu, “Security of voip networks,” in *2010 2nd International Conference on Computer Engineering and Technology*, vol. 3, pp. V3–104, IEEE, 2010.
- [39] D. Butcher, X. Li, and J. Guo, “Security challenge and defense in voip infrastructures,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1152–1162, 2007.
- [40] A. D. Keromytis, “Voice over IP: Risks, threats and vulnerabilities,” in *Proceedings of the Cyber Infrastructure Protection (CIP) Conference*, pp. 223–240, JSTOR, 2009.
- [41] X. Hu, C. Liu, S. Liu, W. You, and Y. Zhao, “Signalling security analysis: Is HTTP/2 secure in 5G core network?,” in *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, IEEE, 2018.

- [42] A. R. Modarressi and R. A. Skoog, “Signaling system no. 7: A tutorial,” *IEEE Communications Magazine*, vol. 28, no. 7, pp. 19–20, 1990.
- [43] G. mobile Suppliers Association *et al.*, “The road to 5G: Drivers, applications, requirements and technical development,” *Global Mobile Suppliers Association*, available on www.gsacom.com, 2015.
- [44] C. Zhang, X. Wen, L. Wang, Z. Lu, and L. Ma, “Performance Evaluation of Candidate Protocol Stack for Service-Based Interfaces in 5G Core Network,” in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, IEEE, 2018.
- [45] Z. Kotulski, T. Nowak, M. Sepczuk, M. Tunia, R. Artych, K. Bocianiak, T. Osko, and J.-P. Wary, “On end-to-end approach for slice isolation in 5G networks. Fundamental challenges,” in *2017 Federated conference on computer science and information systems (FedCSIS)*, pp. 783–792, IEEE, 2017.
- [46] J. Lu, L. Xiao, Z. Tian, M. Zhao, and W. Wang, “5g enhanced service-based core design,” in *2019 28th Wireless and Optical Communications Conference (WOCC)*, pp. 1–5, IEEE, 2019.
- [47] Q. Li, G. Wu, A. Papathanassiou, and U. Mukherjee, “An end-to-end network slicing framework for 5g wireless communication systems,” *arXiv preprint arXiv:1608.00572*, 2016.
- [48] S. A. Kazmi, L. U. Khan, N. H. Tran, and C. S. Hong, *Network Slicing for 5G and Beyond Networks*. Springer, 2019.
- [49] S. Rommer, P. Hedman, M. Olsson, L. Frid, S. Sultana, and C. Mulligan, *5G Core Networks*. Elsevier, 2019.

- [50] R. Harel and S. Babbage, “5G security recommendations Package# 2: Network Slicing,” *NGMN Alliance*, Apr, 2016.
- [51] V. Del Piccolo, A. Amamou, K. Haddadou, and G. Pujolle, “A survey of network isolation solutions for multi-tenant data centers,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2787–2821, 2016.
- [52] A. Manzalini, C. Lin, J. Huang, C. Buyukkoc, M. Bursell, *et al.*, “Towards 5G software-defined ecosystems: Technical challenges, business sustainability and policy issues,” *IEEE SDN White Paper*, 2016.
- [53] G. S. Blair, G. Coulson, P. Robin, and M. Papathomas, “An architecture for next generation middleware,” in *Middleware98*, pp. 191–206, Springer, 1998.
- [54] S. Gutz, A. Story, C. Schlesinger, and N. Foster, “Splendid isolation: A slice abstraction for software-defined networks,” in *Proceedings of the first workshop on Hot topics in software defined networks*, pp. 79–84, 2012.
- [55] X. Zhang, A. Kunz, and S. Schröder, “Overview of 5G security in 3GPP,” in *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 181–186, IEEE, 2017.
- [56] S. Chen, J. Hu, Y. Shi, Y. Peng, J. Fang, R. Zhao, and L. Zhao, “Vehicle-to-everything (V2X) services supported by LTE-based systems and 5G,” *IEEE Communications Standards Magazine*, vol. 1, no. 2, pp. 70–76, 2017.
- [57] T. Taleb and A. Kunz, “Machine type communications in 3GPP networks: potential, challenges, and solutions,” *IEEE Communications Magazine*, vol. 50, no. 3, pp. 178–184, 2012.

- [58] S. Husain, A. Prasad, A. Kunz, A. Papageorgiou, and J. Song, “Recent trends in standards related to the internet of things and machine-to-machine communications,” *Journal of information and communication convergence engineering*, vol. 12, no. 4, pp. 228–236, 2014.
- [59] S.-h. Sun, J.-l. Hu, Y. Peng, X.-m. Pan, L. Zhao, and J.-y. Fang, “Support for vehicle-to-everything services based on LTE,” *IEEE Wireless Communications*, vol. 23, no. 3, pp. 4–8, 2016.
- [60] S. Doumiati, H. Artail, and D. M. Gutierrez-Estevez, “A framework for LTE-A proximity-based device-to-device service registration and discovery,” *Procedia Computer Science*, vol. 34, pp. 87–94, 2014.
- [61] 3GPP, “3GPP System Architecture Evolution (SAE): Security Architecture,” TS 33.401 V.14.2.0, 3GPP, March 2017.
- [62] D. Jiang and L. Delgrossi, “IEEE 802.11 p: Towards an international standard for wireless access in vehicular environments,” in *VTC Spring 2008-IEEE Vehicular Technology Conference*, pp. 2036–2040, IEEE, 2008.
- [63] T. ETSI, “102 225,” *Secured Packet structure for UICC based applications, Rel-6 v6. 0.6*, 2012.
- [64] A. O. Granerud, “Identifying TLS abnormalities in Tor,” Master’s thesis, 2010.
- [65] J. Mattsson, “Overview and Analysis of Overhead Caused by TLS,” *draftmattsson-uta-tls-overhead-01 (work in progress)*, *IETF*, October, pp. 48–52, 2014.

- [66] A. K. Murthy, R. Parthasarathi, and V. Vetrivel, “Security testbed for next generation mobile networks,” in *2020 Third ISEA Conference on Security and Privacy (ISEA-ISAP)*, pp. 122–129, IEEE.
- [67] T.-J. Tan, F.-L. Weng, W.-T. Hu, J.-C. Chen, and C.-Y. Hsieh, “A reliable intelligent routing mechanism in 5G core networks,” in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pp. 1–3, 2020.
- [68] C.-W. Liao, F. J. Lin, and Y. Sato, “Evaluating NFV-enabled network slicing for 5G core,” in *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 401–404, IEEE, 2020.
- [69] Y. Lin, C. Tseng, and M. Wang, “Effects of Transport Network Slicing on 5G Applications. Future Internet 2021, 13, 69,” 2021.
- [70] C. B. Both, J. Borges, L. Gonçalves, C. Nahum, C. Macedo, A. Klautau, and K. Cardoso, “System Intelligence for UAV-Based Mission Critical with Challenging 5G/B5G Connectivity,” *arXiv preprint arXiv:2102.02318*, 2021.
- [71] 3GPP, “5G end to end Key Performance Indicators (KPI) (R16),” TS 28.554 V16.2.0, 3GPP, September 2019.

Appendix A

A.1 Commissioning 5G SBA Slice and Verifying Control Plane Service Operations According to 3GPP

We used docker-compose tool to commission a 5G core along with all the necessary network functions. Docker container is a modern technology that has been recommended for the deployment of virtualization and software-defined network. In the experiments, we adopted service-based architecture to create a 5G core slice. Having the SBI model in place allows the slice components, namely network functions, to produce services to one another. NRF plays a key role in the model. All the network functions register their profile to the NRF through HTTP primitives.

We also should make sure that the SBI scheme follows HTTP. In another word, it dictates network function to exchange the control plane messages over TCP with no TLS. It is crucial in our implementation since we intend to make performance evaluations in both cases.

All the network functions and related features have been written in Go language in Free5GC open source project that can be accessed through <https://github.com/free5gc/free5gc.git>. By default in Free5GC development, we can

create a slice that consists of network functions. In case HTTPS scheme is selected for securing intra-slice communication, they are only limited to secure their connection with two cipher suites. Nonetheless, as discussed in the methodology chapter we are required to conduct the experiments on different selected TLS cipher suites. To achieve this goal we should identify the source files that provides properties that NFs use when become operative. The file is called `server.go` that is located in `/free5gc-src/free5gc/lib/http2-util/server.go`. In the experiments, we added and changed some parts of the `server.go` source file using Golang library as depicted in Figure A.1 that enables network functions to authenticate and secure their connection with a specific cipher suite.

```
package http2_util

import (
    "crypto/tls"
    "fmt"
    "net/http"
    ...<Some parts have been omitted for brevity>

    if preMasterSecretLogPath != "" {
        preMasterSecretFile, err := os.OpenFile(preMasterSecretLogPath, os.O_WRONLY|os.O_CREATE|os.O_TRUNC, 0600)
        if err != nil {
            return server, fmt.Errorf("create pre-master-secret log [%s] fail: %s", preMasterSecretLogPath, err)
        }
        server.TLSConfig = &tls.Config{
            KeyLogWriter: preMasterSecretFile,
            MinVersion:   tls.VersionTLS12,
            MaxVersion:   tls.VersionTLS12,
            CipherSuites: []uint16 {
                tls.TLS_RSA_WITH_3DES_EDE_CBC_SHA,
                //tls.TLS_RSA_WITH_AES_256_GCM_SHA384,
                //tls.TLS_RSA_WITH_AES_128_GCM_SHA256,
                //tls.TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
                //tls.TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
                //tls.TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA,
```

Figure A.1: Calling TLS cipher suites from crypto library of GoLang

In the experiments, it is assumed that a network slice along with its required parameters is pre-defined and on-boarded. Once a slice is designed and instantiated, it is ready to be activated. At this phase the slice along with its NFs are operative and exchange control plane messages needed to manage network functions and deliver any communication services to the slice end users. Figure

While the operation phase of the slice lifecycle was in progress, we capture all the packets that are transmitted between NRF and the rest of the network functions. We further filter out the messages concerned with NF profile registration individually. This aims at two goals. The first goal is to verify that all the network elements communicated their parameters over Nnrf-NFManagement-NFRegister which complies with the standard specifications in [13]. The second goal is to indicate how we measured the time each network function requires to register itself to the NRF.

We repeated the slice creation and activation twenty-five times. This aims to collect enough sample data to analyze and produce reliable results.

A.2 NF Registration With TLS1.2 Cipher Suites

This part aims to have the network functions of a 5G network slice transmitting the application messages over a specific transport layer security cipher suite for instance AES-256-GCM-SHA384. To achieve this it is required to modify the server.go file that contains and defines the mechanisms, like HTTP, HTTPS, TLS version, and TLS cipher suite.

Having the network functions support TLS requires the docker engine to create network functions images with the security ability at the on-boarding and commissioning phase of slice life. Once the images are created, we run docker-compose to make the slice operative.

A.3 Experiment with simultaneous launch of NFs

This section explains why we ran the experiments one NF at a time. When we did the time measurements with all network functions starting at the same time we

noticed that there are discrepancies between the time measurements of each iteration. For instance, it took PCF to complete the registration process around 8, 30, and 60 milliseconds for different iteration. The reason was that when we launched the network functions concurrently they compete with each other over resources at the same time to register their profile with NRF. Figure A.2 depicts the total registration time in millisecond for each network function for four iterations. As you notice the time measurements are skewed.

Figure A.2: There are significant discrepancies in total registration time when NFs register with NRF concurrently.

Therefore, we attempted to let network functions register with NRF one at a time to isolate TLS impact as much as possible to minimize the discrepancies in the measurements. So, one network function registers with NRF multiple times sequentially without having any other network function does the same. Figure A.3 demonstrates the results of four repetitions of the NF registration with the result in a close range.

Figure A.3: The total registration time when the NFs register with NRF one at a time.