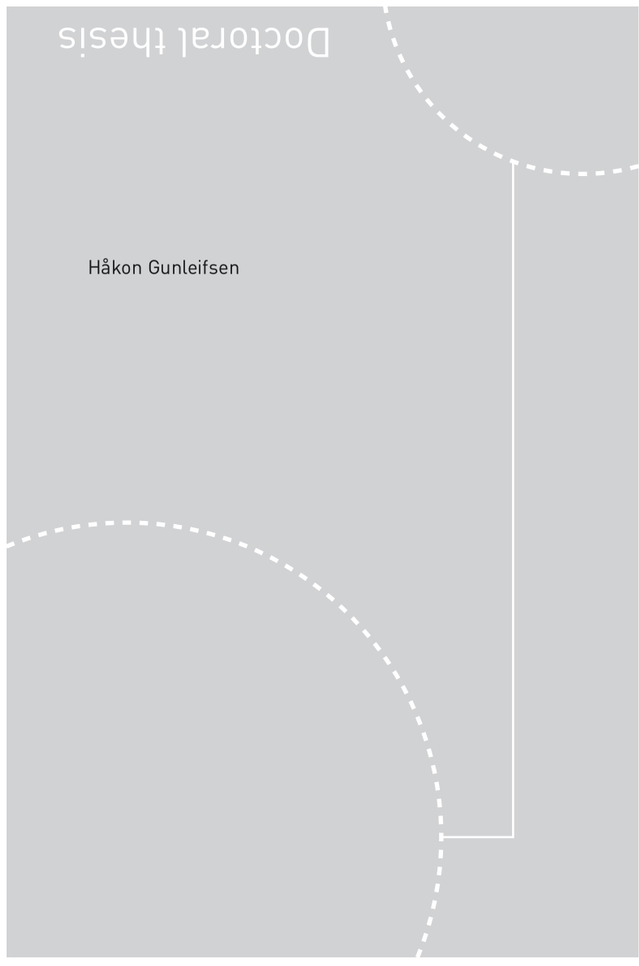
Doctoral theses at NTNU, 2019:388

Håkon Gunleifsen

Security in Interconnected Network Function Virtualisation Environments



ISBN 978-82-326-4402-5 (printed ver.)
ISBN 978-82-326-4403-2 (electronic ver.)
ISSN 1503-8181

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Information Technology and Electrical
Engineering
Department of Information Security and
Communication Technology

Doctoral theses at NTNU, 2019:388

NTNU
Norwegian University of
Science and Technology

Håkon Gunleifsen

Security in Interconnected Network Function Virtualisation Environments

Thesis for the Degree of Philosophiae Doctor

Trondheim, December 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Information Security and Communication Technology

ISBN 978-82-326-4402-5 (printed ver.)
ISBN 978-82-326-4403-2 (electronic ver.)
ISSN 1503-8181

IMT-report 2019:388

Doctoral theses at NTNU, 2019:388

Printed by NTNU Grafisk senter

Security in interconnected Network Function Virtualisation environments

Håkon Gunleifsen

This thesis is submitted to the
Norwegian University of Science and Technology
for the degree of Doctor of Philosophy in
Information Security

NTNU
Norwegian University of
Science and Technology

...there is no prescribed route to follow to arrive at a new idea. You have to make the intuitive leap. But the difference is that once you've made the intuitive leap you have to justify it by filling in the intermediate steps. In my case, it often happens that I have an idea, but then I try to fill in the intermediate steps and find that they don't work, so I have to give it up.

Stephen W. Hawking (8 Jan 1942 - 14 Mar 2018)

Declaration of authorship

I, Håkon Gunleifsen, hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated

Signed:

(Håkon Gunleifsen)

Date:

Abstract

Network Functions Virtualization (NFV) aims to change how network operators handle their network equipment. It also aims to change how end-users shop their network service. NFV is a paradigm shift of networking which consists of moving the physical network appliances from hardware to software. This enables providers to run these network devices in remote data centres. One example of this concept is that end-users do no longer need to have a stack of residential network equipment. They can simply move their network devices to the cloud. This concept of virtualising network equipment has the potential to significantly reduce hardware cost, decrease the time-to-market, expand the lifetime of the network devices and save operational expenses. However, security remains a major concern for operators and end-users before they are willing to adopt the technology more widely. The border security arranged by physical network devices becomes more unclear for the end-users, and they can easily question who has access to their virtual network devices. The concept of virtualisation also enables the virtual network devices to be run at any service provider. Then, this also questions what provider who has access to what data. If all network traffic from the end-users are going through multiple services at multiple providers, then the end-user can question, who has access to what and who can access their data traffic? In fact, the end-users have very little control over this. However, it is obvious that the privacy of the end-users is important. They should be able to know what provider who can access their data traffic, who can access what and whether they share an NFV network service with someone else. They should also be able to know if their homes are protected from cyber-attacks.

Correspondingly, the main objective of this research is to provide a mechanism which ensures the confidentiality, integrity and availability of the end-users' NFV traffic. In particular, it aims to secure end-user communication when Internet Service Providers are sharing virtual network service platforms between each other. This includes protecting the integrity of the data traffic and achieving data traffic confidentiality, which currently is very limited in NFV environments.

The first part of the research contains a study of the security implications of putting

a virtual networking device into the cloud. This research aims to put a focus on the aforementioned research challenge and investigate what security mechanisms which can be used to achieve integrity and confidentiality. This research challenges the current standards and asks whom the end-user can trust in a multi-provider NFV environment. Further, this research results in a set of requirements which must be fulfilled in order to achieve the security objectives.

These security concerns present a major obstacle for NFV adoption. Hence, the second part of the research presents an architecture of how to overcome these security challenges. The focus in these studies concerns how the access control can be achieved by low-level packet isolation and how it can be abstracted to network orchestration policies. The key elements in this research challenge are how to exchange keys and how to steer encrypted data packets.

The last part of the research is related to the development of a framework which supports the confidentiality, integrity and the availability of the data traffic in NFV. Here, this research aimed to verify that the implementation of the architecture fulfils the requirements which were developed in the first part of this research. The final results show that these requirements are fulfilled. In the context of NFV adoption, this research contribution of access control and confidentiality can affect the perspective of security and trust in NFV networks for both end-users and operators. Correspondingly, it can also have an impact on NFV adoption in general.

Sammendrag

Nettverks Funksjons Virtualisering (NFV) tar sikte på å endre hvordan internettleverandører håndterer og selger nettverksutstyr. Det tar også sikte på å endre hvordan sluttbrukere bruker og kjøper sine nettverkstjenester. NFV paradigmeskiftet består av konseptet med å flytte fysiske nettverksbokser ifra hardware til software. Dette gjør det mulig for nettverksoperatører å kjøre disse tjeneste i eksterne datasentre. Et eksempel på dette er at sluttbrukere ikke lenger trenger å ha en stabel med nettverksbokser stående hjemme. De kan enkelt fortalt bare flytte disse ut i skya. Dette konseptet med å virtualisere nettverksutstyr kan medføre en signifikant endring av utstyrskost, leveransetid, levetid på boksene og det kan også redusere logistikkkostnadene. Likevel så er sikkerheten en av de største bekymringene for både sluttbrukere og nettverksleverandører. Dårlige sikkerhetsløsninger er et hinder for at denne teknologien blir tatt i bruk. Disse boksene som tidligere stod for en datasikkerhetsbarriere i hjemmene, blir nå liggende i skya. Disse sikkerhetsskillene blir da ikke lenger like enkle å forhold seg til. Når boksen ligger i skya, så kan det også stilles spørsmålstegn til hvem som faktisk har tilgang til datatrafikken. Teknologien åpner også for at disse boksene kan ligge hos flere operatører og da blir det enda vanskeligere å finne ut hvem som faktisk har tilgang til hva og kunne vite hvem som kan faktisk se datatrafikken. Faktisk har sluttbrukere veldig liten kontroll på dette. Det hevdes at dette er en viktig sak for alle sluttbrukere i forhold til at de har kontroll på sine egne data. De bør kunne vite hvilke skyleverandører som kan avlytte datatrafikken deres, hvem som kan avlytte hva og hvorvidt de deler en nettverkstjeneste med andre. De bør også kunne vite hvordan hjemmene deres er beskyttet mot dataangrep.

Hovedmålet vårt med denne forskningen er å utvikle en mekanisme som sikrer sluttbrukerne mot disse truslene. Datatrafikken må sikres. Spesielt viktig er dette når ulike skyleverandører har tilgang til den samme datatrafikken. Teknologien er i dag slik at alle involverte skyleverandører faktisk har tilgang til alt. Derfor må løsningen kunne sikre at sluttbrukeren selv kan bestemme hvem som skal få lov til å se de ulike bitene av datatrafikken. Integriteten av datapakkene må beskyttes og innholdet i dem må krypteres. Hvis datapakkene er kryptert, så kan brukeren

selv bestemme hvilken leverandør som skal kunne behandle de ulike bitene av datatrafikken. Nettverksfunksjonsvirtualiseringsteknologien som finnes i dag har svært liten støtte for dette.

Den første delen av forskningen inneholder en studie av sikkerhetsutfordringene rundt dette. Her tar en sikte på å fremheve sikkerhetsutfordringene og undersøke hvilke muligheter som finnes for å gjøre noe med dem. Dette spesielt med tanke på å innføre tilgangskontroll og kryptering. Denne forskningen utfordrer dagens standard og stiller spørsmåltegn til hvem sluttbrukerne egentlig skal forholde seg til når en skal gjøre en slik sikring mot flere skyleverandører. Et av disse bidragene i denne forskningen viser at det må stilles en rekke overordna krav som må oppfylles for at dette kan sikres.

Sikkerhetsutfordringene utgjør et betydelig hinder for at NFV skal bli tatt i bruk. Derfor er vårt andre forskningsbidrag en løsningsskisse for hvordan dette kan løses. Fokuset i disse studiene har handlet om hvordan vi kan klare å få separert datatrafikken og hvordan en sluttbruker kan forholde seg til dette. De viktigste elementene i denne utfordringen ligger i hvordan vi på en sikker måte kan utveksle nøkler mellom tjenesteleverandørene og hvordan en faktisk kan klassifisere krypterte datapakker.

Den siste delen i forskningen vår handler om å lage et rammeverk og en demonstrasjon av en løsning som støtter denne integriteten og konfidensialiteten til datapakkene i NFV. Her ønskes det å kvalitetssikre denne implementasjon mot kravene som ble stilt i første delen av forskningen. Sluttresultatet viste at disse kravene ble oppfylt. I forhold til at NFV teknologien skal bli tatt videre i bruk, så anses dette som et viktig bidrag, da dette påvirker tilliten til NFV for både nettverksleverandører og sluttbrukere.

Acknowledgements

It is a pleasure to thank all those who helped me bringing this thesis to life, with their support, patience, and advice.

First and foremost, I would like to thank my life partner and family for always being the greatest support to all my endeavours, with understanding, encouragement and helpful advice in any misfortune. Similarly, I would like to thank the CEO of Eidsiva bredbånd, Trond Skjellerud, who gave me the opportunity to pursue my PhD and combining it with working 25% in Eidsiva bredbånd. He enabled a great collaboration and the financial support from Eidsiva, the Center for Cyber and Information Security, the Norwegian Research Council and NTNU.

I would like to express my gratitude to my main supervisor Prof. Thomas Kemmerich for being a leading example of commitment and great guidance. His insightful advice allowed me to identify the true meaning of research and academic work. My co-supervisor Prof. Slobodan Petrovic has also been very helpful to my work and inspired me to explore additional research areas of my interest. A special thanks goes to my colleagues in the Critical Infrastructure and Resilience group at NTNU. In particular, I would like to thank Dr Vasileios Gkioulos who has been my closest co-worker and friend during my studies. His expertise, encouragement and guidance will always be greatly indebted.

A special gratitude also goes to my supervisor in Eidsiva bredbånd, Tore Baarstad. In addition to his helpful advice, he early taught me always to question and challenge the incorporated truths.

Furthermore, I would like to thank all members of the COINS Research School of Computer and Information Security, Faculty of Information Technology and Electrical Engineering and the Center for Cyber and Information Security for creating an excellent working environment. It has been a pleasure and an honour to work within such teams with a cooperative spirit and fruitful discussions

Moreover, I would like to express my gratitude to my friends, whose company and encouragement has been a great support. Last but not least, I would like to thank

the administrative staff and my colleagues in both Eidsiva bredbånd and at NTNU for great support and cooperation.

Contents

List of Tables

List of Figures

List of Abbreviations

AAA	Authentication, Authorization, and Accounting
AFI	Address Family Identifier
ALTO	Application Level Traffic Optimization
API	Application Programming Interface
ARP	Address Resolution Protocol
AS	Autonomous system
AuC	Authentication Center
BGP	Border Gateway Protocol
BNG	Broadband Network Gateway
BMv2	Behavioural Model 2
BSS	Business System Support
CDNi	Content Distribution Network Interconnection
CF	Classification Functions
CHAP	Challenge-Handshake Authentication Protocol
CLI	Command Line Interface
COP	Control Orchestration Protocol
CPE	Customer Premise Equipment
DSCP	Differentiated Services Code Point
DSRM	Design Science Research Methodology
EAP	Extensible Authentication Protocol
EF	Encryption Function aka Encrypting Service Function
EL	Encrypted Link
EM	Element Managers
ETSI	European Telecommunications Standards Institute
EVNF	Encrypted Virtual Network Function
FIB	Forwarding Information Base
FORCES	Forwarding and Control Element Separation protocol
GDPR	General Data Protection Regulation
GET	Group Encrypted Transport
GENEVE	Generic Network Virtualization Encapsulation

gNMI	Google Remote Procedure Call (gRPC) Network Management Interface
gNOI	Google Remote Procedure Call (gRPC) Node Operator Interface
GRE	Generic Route Encapsulation
HCI	Hyper-Converged Infrastructure
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
IaaS	Infrastructure as a Service
I2NSF	Interface to Network Security Function
I2RS	Interface to the Routing System
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
IKE	Internet Security Key Exchange protocol
IP	Internet Protocol
IPsec	Internet Protocol Security
IPVPN	IP Virtual Private Network
IRTF	Internet Research Task Force
ISP	Internet Service Provider
KINK	Kerberized Internet Negotiation of Keys
KMS	Key Management Services/System
KVM	Kernel-based Virtual Machine
L2TP	Layer 2 Tunneling Protocol
LISP	Locator Identifier Separation Protocol
LLDP	Link Layer Discovery Protocol
LSO	Life Cycle Orchestration
MANO	Management and Orchestration
mBGP	multi-protocol BGP
MdO	Multi-Domain network Orchestration
MEF	Metro Ethernet Forum
MPLS	Multi-protocol Label Switching
MPLS-SR	Multi-protocol Label Switching with Segment Routing
MSCHAP	Microsoft Challenge-Handshake Authentication Protocol
MTU	Maximum Transfer Unit
NBI	Northbound Interface
NCR	Network Controller Route
NETconf	Network Configuration Protocol
NF	Network Function (aka VNF aka SF)
NGMN	Next Generation Mobile Networks

NS	Network Service
NFV	Network Function Virtualisation
NFVI	Network Function Virtualisation Infrastructure
NFVIaaS	NFV Infrastructure as a Service
NSD	Network Service Descriptors
NSH	Network Service Header
NVGRE	Network Virtualisation using Generic Route Encapsulation
NVI	New Virtual Interfaces
NVO	Network Virtualisation Overlay
ODL	OpenDayLight
OF	OpenFlow
ONF	Open Networking Forum
OPNNFV	Open Network Function Virtualization
OSS	Operation Support System
OVS	Open Virtualised Switch
P4	Programming Protocol-Independent Packet Processors
PAD	Peer Authorization Database
PaaS	Platform as a Service
PAP	Password Authentication Protocol
PBR	Policy based Routing
PCE	Path Computation Element
PCEP	Path Computation Element Protocol
PCRF	Policy and Charging Rules Function
PEAP	Protected Extensible Authentication Protocol
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PSK	Preshared Key
PGW	Packet Data Network Gateway
RFC	Request For Comments
RO	Resource Orchestration
RPKI	Resource Public Key Infrastructure
RR	Route Reflector
SA	Security Association
SaaS	Software as a Service
SAD	Security Association database
SAFI	Subsequent Address Family Identifier
SAML	Security Assertion Markup Language
SASL	Simple Authentication and Security Layer

SBI	Southbound Interface
SC	Service Chain aka SFC
SD-IKE	Software-Defined Internet Key Exchange
SD-SA	Software-Defined Security Associations
SDDC	Software-Defined Data Centre
SDN	Software-Defined Network
SDNi	Software-Defined Network interface
SDNRG	Software-Defined Networking Research Group
SDR	Software-Defined Radio
SDWAN	Software-Defined Wide Area Network
SF	Service Function
SFC	Service Function Chain
SFF	Service Function Forwarders
SFI	Service Function Identifier
SFIR	Service Function Instantiated Routes
SFIR-E	Service Function Instantiated Routes with Encryption Service
SFP	Service Function Path
SFPR-E-RD	Service Function Path Routes with Encryption Route Distinguisher
SFPR-RD	Service Function Path Routes with Route Distinguisher
SI	Service Index
SID	TMforum Information Framework
SO	Service Orchestration
soBGP	Secure origin Border Gateway Protocol
SON	Self-Organising Networks
SPD	Security Policy Database
SPI	Service Path Index
SPRING	Source Packet Routing in Networking
SR	Segment Routing
STT	Stateless Transport Tunnelling Protocol
TE	Traffic Engineering
TL	Transport Link
TLV	Type, Length, Value
TMF	Telecommunication Management Forum
TOSCA	Topology and Orchestration Specification for Cloud Applications
TPM	Trusted Platform Module
TR	Transport-link Route
vCPE	virtual Customer Premise Equipment

vFW	Virtual Firewall
VIM	Virtual Infrastructure Manage
VL	Virtual Links
VLD	Virtual Link Descriptor
VM	Virtual Machine
VNF	Virtual Network Function
VNFaaS	VNF as a Service
VNFC	Virtual Network Function Components
VNFFGD	VNF Forwarding Graph Descriptor
VNFM	Virtual Network Function Manager
VNPaaS	Virtual Network Platform as a Service
VOIP	Voice Over Internet Protocol
VPN	Virtual Private Network
VPP	Vector Packet Processing
VTEP	Virtual eXtensible Local Area Network Tunnel End Point
VXLAN	Virtual eXtensible Local Area Networks
VXLAN-GPE	Virtual eXtensible Local Area Network Generic Protocol Extension
YANG	Yet Another Next Generation (model for NETconf)

Part I

PhD thesis

Chapter 1

Introduction

1.1 Motivation

In contemporary operator networks, the network appliances are populated with a large and increasing variety of proprietary hardware. Launching a new network service often requires yet another hardware device where the accommodation of space, power and the integration of a new management tool is becoming increasingly difficult. Additionally, increasing energy costs, integration costs, new staff skills and relatively low hardware life-cycles, constraint innovations and capitalisation among network service providers.

Network Function Virtualisation (NFV) aims to address these problems by leveraging contemporary virtualisation technologies. Running virtual network appliances in a data centre consolidates hardware types, reduces time-to-market, reduces hardware failures and enables a new set of eco-systems and business models. One of these new eco-systems is related to sharing data centre resources. The NFV technology enables the virtualised network equipment to be run in multiple data centres or distributed nodes. Network operators can choose to fully or partly outsource their data centre resources and dynamically move the network services to the most cost-efficient data centre. This dynamic deployment of NFV across multiple domains necessitates a secure interconnection method between the operators (Figure: 1.1). However, the current NFV networking standards have limitations regarding protecting the privacy of the end-users, and it has general security vulnerabilities [1]. Hence, the main research motivation is to increase the security level of the NFV networking standards in order to make NFV more adoptable to the market and for the end-users.

However, six years after the first NFV standardisation effort from the European Telecommunications Standards Institute (ETSI), Whitestack [2] claims that the

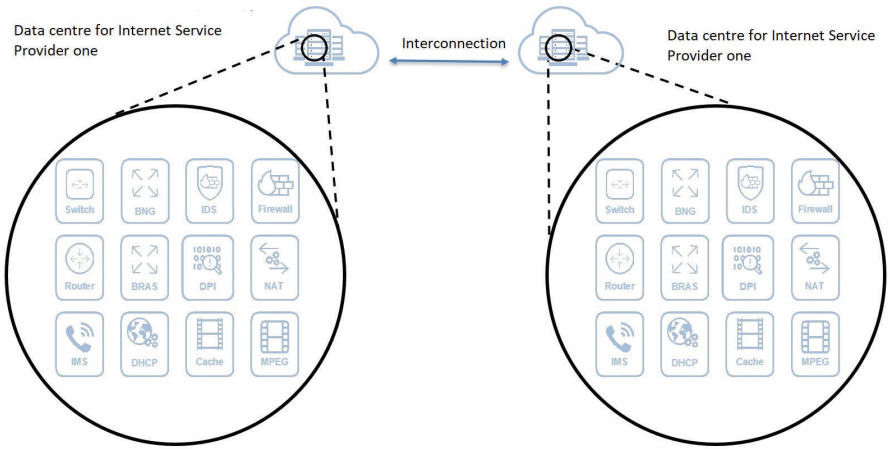


Figure 1.1: Interconnected NFV domains

adoption rate of NFV is slow and that the operators are deploying NFV in a vertical approach. A vertical approach implies that each operator is deploying their own NFV stack without utilising multiple data centre clouds. Whitestack also claims that it is the vendors and not the operators who are driving the NFV innovations [2]. Operators tend to trust their integrators and vendors, where they buy standalone NFV solutions from them in the same way they previously bought hardware appliances. As a result, the operators are ending up with multiple NFV stacks internally in their data centre, which are not interconnected operationally. These different silos of NFV environments potentially result in more expensive deployments. This research questions why operators do not seek horizontal interconnections of the data centres as the technology promises. The current research has shown that there is still a lack of security features in the solutions, and there is a variety of different actors competing for setting the NFV standards. It is assumed that this lack of standardisation and the lack of trust to NFV security functions is a signal of immaturity of NFV, which prohibits operators from embracing the technology.

Eidsiva bredbånd is a Norwegian Internet Service Provider and a funding partner in this research. They aim to tackle this problem by gaining more knowledge about NFV, deploying multi-data centre solutions and fully automate service provisioning across service providers in a secure manner. They perceive that a part of the problem is that the operators are stuck in the old business models and that the service providers lack trust between each other for these virtualised network services. One motivation behind the funding of this industrial PhD is, therefore, to investigate the security threats of interconnected NFV environments and potentially sug-

gest a solution to overcome the problem. These business drivers correlate with the academic drivers of increasing the level of NFV security. A better understanding of the security threats is also expected to raise a greater security awareness among both operators and end-users, which potentially can increase the NFV adoption rate and set new service demands from end-users.

Eidsiva bredbånd is a regional Internet Service Provider (ISP), which similar to most other ISPs has a footprint focused on a specific region. This results in a market fragmentation, which makes it challenging to deploy cost-effective infrastructure services across these regions or between countries. Even if most ISPs deliver similar services, the inter-operator tools for collaborating in service delivery are limited. This makes inter-ISP service delivery very time-consuming. Similarly, historical ISP business merging and technology evolution have resulted in a technology fragmentation internally in many of these companies [3]. This fragmentation within operators organisations consists of multiple network domains, multiple equipment vendors and multiple technologies (i.e. optical and coax). They often have different tools of management and different groups of staff specialised on a subset of these fragments. Service delivery often requires cooperation across these autonomous fragments where multiple operational groups represent a major bottleneck for the service delivery speed in these companies. Hence, this represents a need for securely interconnecting these domains in order to provision services more efficiently. This applies both between operators and within one operator domain. Here, standardising interfaces by using NFV is a key enabler.

Seen from the end-user perspective, interconnecting service providers is also a motivational factor for the society at large. A competitively market most frequently benefits the end-users. Due to the regional footprint of many network infrastructures, it can be claimed that the end-users are currently, directly or indirectly, enforced to buy their main network services from their ISP. This applies to both services, infrastructure and network security. One example of this problem is how governmental regulations are enforcing network infrastructure owners to let other operators operate the last-mile infrastructure to the end-user [4]. This is referred to as open network models. In many open networks, changing the ISP is a time-consuming process. Following the principles of virtualised networks, it is possible to share the virtual infrastructure instead of the physical network. Then, the end-user can change services on-demand across multiple ISPs. However, modern networks are highly integrated with computational resources in data centres and many types of infrastructure networks (i.e. access, distribution and core networks). Opening up a network to be operated by multiple network and data centre service providers is potentially a security challenge for both the end-users and the operators. According to the General Data Protection Regulation (GDPR) [5],

the end-users require to know about operators having access to their private data, which implicitly also includes who has access to monitor their internet traffic. An interconnected network, which includes network services, exposes the end-users' identity and their usage patterns to multiple providers. Similarly, enterprises using multi-cloud based network services, requires that cloud service provider one does not have access to information which is directed to cloud service provider two.

From the network operator perspective, they need to have control over their on-premise physical network as well as of their off-premise virtual networks within remote infrastructures. Letting multiple operators control the network configuration, calls for more sophisticated security policies. Hence, it is a fundamental premise for future network interconnection that horizontal interconnections between network operators tenants are securely modelled and standardised in an automated manner. However, such interconnection frameworks are also a security threat themselves.

Correspondingly, this research aims to contribute to ensuring end-user confidentiality and integrity in NFV by securing the NFV data traffic across multiple NFV infrastructures.

1.2 Aim and Scope

Motivated by the aforementioned business and end-user drivers, this research aims to utilise the NFV technology to provide secure and flexible virtual network services for the end-users in multi-operator networks. In particular, it aims to secure end-user communication when ISPs are sharing virtual network services between each other. This includes protecting the integrity of the data packets and maintaining the availability of the network services.

In a general NFV context, ETSI has outlined different areas of research problems for NFV [6]. The scope of this research is limited to a subset of the general NFV research problems. In the ETSI model (Figure: 1.2), the research focus is put around the networking parts of NFV, which this research refers to as the Network Infrastructure Domain. Hence, this research scope partly correlates with a subset of the research problems defined by ETSI [7]. In one of the first NFV document released by ETSI, they defined a set of use cases for NFV [7]. This document describes different methods of both utilising NFV services and interconnecting NFV domains from a high-level service perspective. In the first three NFV use cases [7], ETSI describes how the different components in the NFV model can be interconnected and accessed across different operators. In these use cases, they defined three main service models, which reflects the most relevant interconnection topologies. These relevant models are named: NFV infrastructure as a Service

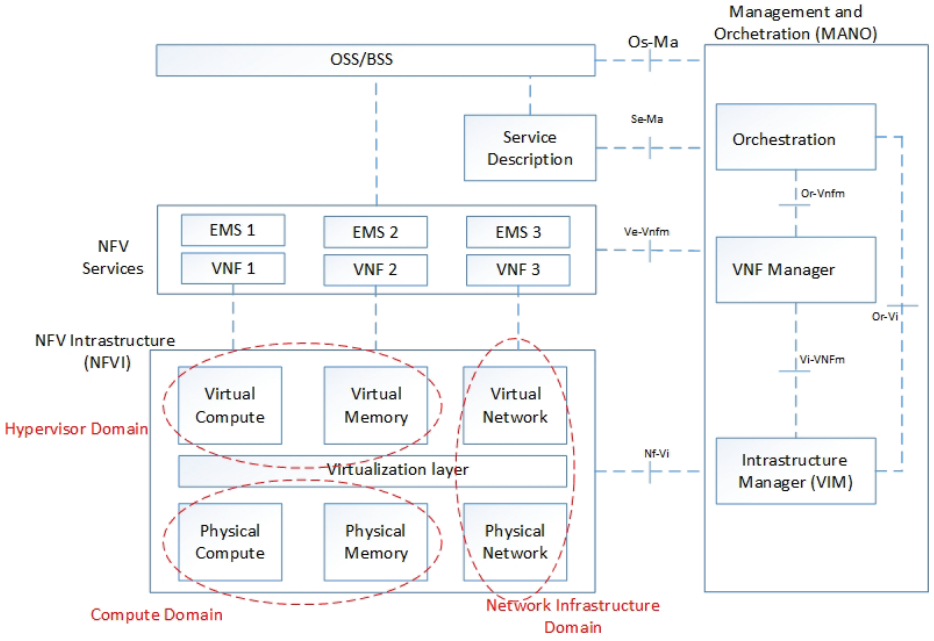


Figure 1.2: ETSI model of NFV

(NFVIaaS), Virtual Network Function as a Service (VNFaaS) and Virtual Network Platform as a Service (VNPaas). For each of these models, ETSI state that: (1) "The NFVI should provide mechanisms such that VNF instances can only access the physical and virtual network terminations to which their access is authorised." (2) "The virtualised environment needs to guarantee complete isolation among users." (3) "The infrastructure resources need to provide mechanisms to separate workload from different operators". Within this study, the focus is set on NFVIaaS, which from a network perspective is perceived to include the other two use cases. Accordingly, these problem statements are consolidated, and it is hereby stated that there is a need for access control and confidentiality in the underlying NFV infrastructure.

Securing communication networks, particularly virtualised networks, is a complex problem which consists of several interlaced work tasks. ETSI has developed a security problem statement and a threat model [8] for NFV security in general. They categorised the different work tasks of NFV security which includes work items such as privacy concerns, regulatory concerns, multi-layer administration, sensitive components, management, orchestration and component-specific security specification. This study seeks to further investigate one specific aspect of these

security problems. This security problem is related to the multi-layer administration of the different interconnection methods between NFV domains and how this concerns end-user privacy [8].

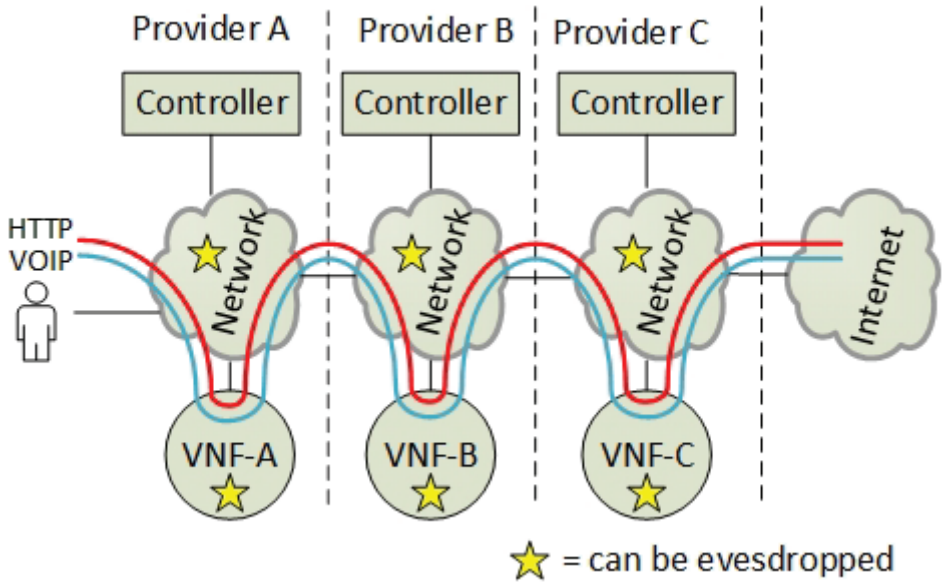


Figure 1.3: Adversary model of NFV packet security

Focusing on the aforementioned subsets of problem statements in the NFV domain defined by ETSI, this research consolidated the problem and developed an adversary model which reflects the scope of this thesis. Figure 1.3 shows a use case of three inter-connected ISPs which collaborate in providing virtual network services for an end-user. The traffic flow from an end-user, which is traversing multiple domains, is sensitive to packet eavesdropping or packet modification both in the NFV network and in the Virtual Network Functions (VNFs). Hence, this research aims to investigate how packet confidentiality and packet integrity can be provided for the end-users in such scenarios. This indicates that there is a need for a network protocol which can protect the network packets from being eavesdropped or modified. Further, it also questions how the security aspects of confidentiality, integrity, resilience and interoperability can be deployed across multiple data centres or ISP domains.

This research limits the use-case to include fixed underlying infrastructures only, primarily based on IP over Ethernet. Mobile services, optical network and satellite communication systems are not the main focus of this work. From a network service type perspective, the research focus is put on end-user network services,

which from a network operator point of view, includes providing network functions for enterprise users, business users and residential consumers.

1.3 Research Objectives

The aforementioned scope of this research is to contribute to providing automated solutions for network confidentiality, integrity and availability in interconnected NFV environments.

The integrity concern questions how we can make sure that an end-user data packet is (1) not tampered with, (2) not redirected by malicious interceptors and (3) that it is not a victim of manipulated forwarding tables. These are attack models which can be used to compromise the high-level security and network service specification. As a result of this, the attacker can potentially bypass network security functions which are protecting an end-user.

The confidentiality concern is related to eavesdropping from malicious network service providers, malicious service functions or other third party eavesdroppers. Specifically, the eavesdropping vulnerability questions why there is no authorisation of the end-user traffic going towards the network services. In this context, this implies a need for isolation and encryption of the end-user traffic.

Hence, the main objective of this research is to provide a mechanism that ensures the integrity and confidentiality of the end-users' NFV traffic in particular. Additionally, this research questions how the availability of the security functions can be maintained during hardware failures and network service migrations.

In order to achieve this objective systematically, the theme of the main objective is further divided into consecutive research questions. Consequently, this research builds upon five steps which reflects the aspects which this research aims to investigate. The steps include (1) identifying the research gap, (2) defining the requirements for a solution, (3) suggesting an architecture, (4) implementing a solution and (5) verify that the architecture works. Based on these objectives, it is defined a set of research questions which reflect this.

1.4 List of Research Questions

This study was driven by the research questions listed below, which have been formulated in alignment with the aforementioned motivation, scope and the top-level research objective.

- **Research Question 1 (RQ-1):** What are the existing NFV interconnection methods, what are the security gaps within them and what is the most secure,

and promising network protocol for interconnected NFV-domains?

- **Research Question 2 (RQ-2):** What are the security requirements and constraints when interconnecting virtual network services between Internet Service Providers.

- **Research Question 3 (RQ-3):** Which are the required architectural components and functionalities, for the enforcement of security control within interconnected network service infrastructures?

- **Research Question 4 (RQ-4):** How can a security control implementation ensure that the end-user privacy is protected by the network service infrastructures?

- **Research Question 5 (RQ-5):** Given the results of the previous research questions, where a suitable reference architecture is developed, how can this secure architecture be deployed and adopted within the current virtualised infrastructures and how does the deployment satisfy the security requirements?

1.5 Research Method

This research is following the Design Science Research Methodology (DSRM) defined by Peffers et al. [9] as it provides a comprehensive guideline to perform scientific research. The method starts with identifying the problem and the motivation. This is followed by abducing the solution's objective. In the next phase, an artefact is created, demonstrated and evaluated. This is performed deductively. Finally, the result of these phases is communicated [10] (Figure: 1.4).

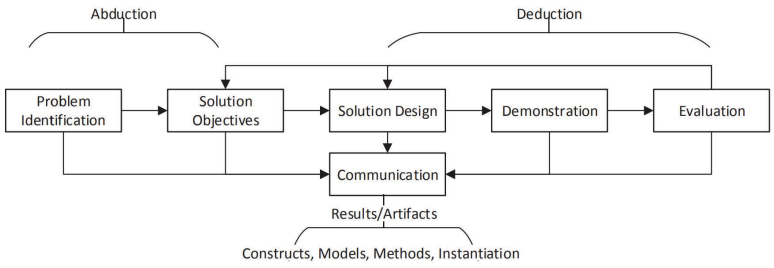


Figure 1.4: The Design Science Research Methodology [9]

The DSRM process is iteratively performed and can be initiated at any stage. This is reflected in this research, where the deductive steps are performed iteratively.

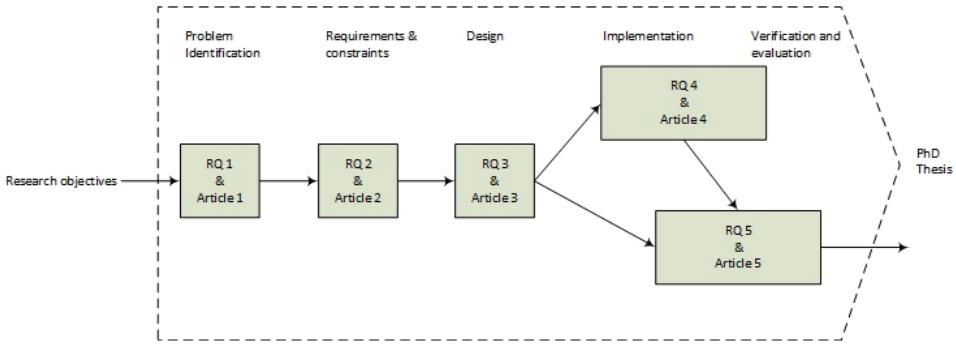


Figure 1.5: The DSRM connection in the Research questions

First (RQ-1), the operational constraints for NFV interconnection and forwarding standards were surveyed. This was driven by the motivation for this research.

Second (RQ-2), this research aimed to define the security requirements and constraints for a new solution based on results from research question 1.

Thirdly (RQ-3), a new architecture was developed, aiming to accommodate the requirements and constraints.

The fourth step (RQ-4) in these studies was to develop a security protocol for exchanging encryption keys between Network Services based on the developed architecture. According to the DSRM, this is an iterative step which also includes design, implementation and evaluation.

The final step in these studies (RQ-5) integrates the previous results into a customised NFV environment for a proof-of-concept verification. Accordingly, this research aimed to verify that the implementation fulfils the requirements which was developed in the second step. Following the iterative DSRM, it is here also possible to make architectural changes from the previous steps. This method is reflected in five consecutive published articles (Figure: 1.5).

1.6 Dissertation Structure

This dissertation is organised into two parts. Part I describes the thesis overview, while part II consists of five research publications.

Chapter 1 introduces the thesis by giving the motivation for the research together with the scope and research objectives. It also describes the research method. Chapter 2 gives background information to the research contributions, while chapter 3 presents the details of the related work. Chapter 4 provides a summary of the publications and also summarises the total contribution of this thesis. Limitations

and future work are discussed in Chapter 5, while Chapter 6 concludes this thesis.

Chapter 2

Background

This thesis tackles the challenges of providing network confidentiality in NFV when interconnecting ISPs' virtual infrastructures. This chapter gives an extensive background from how the virtualisation technologies emerged from different virtualisation technologies to how it ended up in the aforementioned research challenges. It starts by introducing cloud computing and how virtualisation can save costs and increase network security in cloud infrastructures (Section: 2.1). Further, it explains how Software-Defined Networks (SDN) and virtualised networks are key-enablers for network control and distributed network security. NFV is a sum of the elements from these technologies (Section: 2.1.2, 2.1.3). Hence, this chapter also gives a background to NFV and why the traffic steering in NFV has some fundamental security concerns related to integrity and confidentiality (Section: 2.2). Section 2.3 shows how the lack of integrity checks can make attackers bypass security functions in NFV and how the lack of confidentiality makes the data traffic vulnerable for eavesdropping. Further, this chapter presents the existing frameworks which can be used to tackle these research challenges (Section: 2.5). The contributions to this research challenge are focused on interconnecting multi-cloud environments. Hence, the background of the technology trends in the market is also presented (Section: 2.6). This is followed by a technology-specific section which describes one of the most important tools in this research - P4 (Section: 2.7). The final part of this chapter (Section: 2.8) summarises all the background information.

2.1 Virtualisation technologies

During the last decade, virtualisation technologies such as cloud computing, network virtualisation, SDN and NFV have gained attention. The technologies are different, but in some aspect, they are overlapping. From a historical perspective, disruptive technologies are a result of a need for a more flexible service deploy-

ment and a need for a more efficient operational environment of both compute and network resources.

2.1.1 Cloud computing

In 2001, cloud computing was gaining attention. VMware was the first company which fully commercialised server virtualisation, where they enabled multiple operating systems to run on one physical server [11]. The virtualisation technology solved the problem of over-provisioned servers. For example, in 2002, Amazon claimed that they were only using 10% of their compute resources [12]. Their cloud computing infrastructure model [13] solved this problem and it also enabled them to sell and provision on-demand virtual servers from a webpage.

Since then, the concept of cloud computing has evolved to include more than provisioning virtual servers with operating systems. However, the term cloud computing is still associated with on-demand provisioning of the server-side services [14]. Nowadays, cloud computing is a paradigm and a vision of computing as a utility, which makes server-side software available as on-demand services. Cloud computing is, therefore, mostly associated as a platform for operators and developers with innovative ideas. Instantiating a service in the cloud does no longer require a large capital outlay, personnel expenses to operate it or long provisioning times. Innovative companies can get their server-side services up and running quickly without concerning about scaling or costs.

From a service perspective, cloud computing has traditionally used three main categories of services [15]:

- IaaS (Infrastructure-as-a-Service) is a concept which historically provides access to hardware, storage, servers and data centre space or network components. However, a common way of defining this service is that it gives the consumer access to the hypervisor on physical servers. This hypervisor access can either be shared with others (tenants) or the consumer could have exclusive rights to the hardware.

- PaaS (Platform-as-a-Service) is a service where the provider operates the operating system and most frequently also the server-side application. In PaaS services, the consumer is provided access to a software platform where software code or a software configuration can be deployed on the service. This is most frequently associated with enabling development frameworks such as .NET, Ruby or PHP for software developers, where the developers easily can deploy their code.

- SaaS (Software-as-a-Service) simply provides a software application for the

consumer. The consumers do not have access to the operating system or the software application running on the server. The consumer only consumes the application such as Google Apps, Salesforce or Dropbox.

From the service producer perspective, these different types of services require automated provisioning of the underlying compute, storage and network resources. Full automation and orchestration of these services are referred to as a Software-Defined Data Centre (SDDC). Orchestrating these underlying services has historically been very complex. Hence, the concept of Hyper-Converged Infrastructures (HCI) emerged in 2012 [16], based on the concept of putting storage, network and compute into one physical device (i.e. Nutanix [17] or OpenStack [18]). This enables data centre operators to run all resources in software and scale the data centre resources quickly by simply adding or removing HCI servers.

In this research, the focus in particular put on the networking part of this provisioning. In an IaaS platform, the consumer is often given the opportunity to set up and configure the network between the server-side services themselves. On the other hand, for PaaS infrastructures, the service producer controls the network. These two distinct services set a clear differentiation of where the responsibility of the network interconnections relies and consequently also the placement of network security control.

Since 2014, cloud security has become a fast-growing service and now provides a security protection level which is comparable to traditional IT security systems. This includes the protection of critical information such as data-leakage and accidental deletion of data and services. However, security, in general, is still a primary concern for operators and enterprises which move their services to the cloud [19]. The latest security challenge for enterprise operators is multi-cloud operations. The concept of multi-cloud implies that an operator or an enterprise uses multiple cloud services in a single heterogeneous architecture. Multi-clouds differs from the concept of hybrid clouds. Hybrid clouds typically integrate a similar deployment model (i.e. IaaS) to both a public infrastructure (i.e. Amazon AWS) and a private infrastructure (i.e. VMWare NSX). A multi-cloud infrastructure implies the use of multiple public clouds (i.e. Amazon AWS, Microsoft Azure) and multiple private clouds with a combination of different deployment models (Iaas, Paas, SaaS). One of the main objectives of using a multi-cloud environment is that there is a root system (inter-cloud) that controls all the other clouds in order to enable one-stop shopping. Then, the consumer can control the costs and control the operations from one single point.

Multi-clouds bring new security challenges to the table, such as data locality and data access. However, first and foremost, the main challenge is network security.

This includes firewalling each service (Section: 2.1.1), provide network encryption across multiple domains and operate different types of access networks for different kind of services. This introduces a need for configuring the network services dynamically both within a data centre cloud and across multiple data centre clouds.

Micro-segmentation and segregation

Network segmentation has historically been a ground pillar in network design. First of all, it is the foundation of IP subnetting, routing and network efficiency. However, the concept is also important with respect to network security. In a security context, segmentation also called "firewall zoning", is a method of dividing a network into different security zones with different access levels. Historically, a good security practise has been to have many security zones with small subnets on the firewall. However, this has always been a compromise between available resources and the level of security.

Micro-segmentation consists of segmenting the security zones down to very small parts, preferably one host per segment. This is achieved by virtually distributing the firewall rules to every access port on physical or virtual switches. SDN is a technology that enables such infrastructure (Section: 2.1.2). However, the concept originated from cloud computing. Here, the distributed virtual firewall is associated with attaching a virtual firewall to each interface of the virtual machines, which abstracts the firewall concept from the underlying SDN technology.

In cloud computing, this concept has resulted in a new security paradigm and new way in perceiving network security. Abstraction layers, virtualisation technologies and centralised control enables automation of security policies in a new way. Historically, firewall operators created firewall rules based on "zoning interfaces" and packet header attributes only. Now, managing a micro-segmented infrastructure enables operators to deploy firewall rules on abstract policies such as the name of the virtual machine. Hence, the security policies are easier to group and manage. Another advantage of micro-segmentation is that there is no longer need for tenant segregation (i.e. MPLS VPNs or 802.1q VLANs). This is because the segregation has implicitly been enforced by overlay networks and the distributed firewall paradigm. In a networking perspective, this has resulted in network designs which depreciate small IP segments and moves towards designing networks in large layer two domains in the data centres.

Correspondingly, the term micro-segmentation has evolved into something which includes more than segmenting a network into smaller parts. It includes developing and enforcing rulesets for controlling the communications between specific

services or hosts. In extent, it is also a matter of definition if the isolation of the network traffic in micro-segmented networks also should include encryption.

Since the nineties, encrypted channels by the Internet Protocol Security (IPSec) [20] has been the primary technology of setting up secure channels between network equipment. Firewall operators have for many years used IPsec between the firewall and other services in order to achieve packet confidentiality and integrity. The underlying SDN technology which enables distributed firewalling lacks this encryption feature. It requires encryption and keying capabilities in the virtual switches. This has created a research gap in both cloud computing and NFV, where the packet confidentiality inside the data centres and between tenants requires attention.

Currently, the packet confidentiality is protected by outer encryption channels between data centres by the use of site-to-site IPsec channels. However, when sharing services across data centres, it is not only the data centre that must be protected from the outside world. Internally, both the users and the services, require packet isolation and confidentiality. This calls for an encrypted channel per service. Operating micro-segmentation across multiple service providers and multiple IaaSs are challenging if two IaaS platforms run different network technologies. There are primarily two approaches to solve this problem. (1) Having a top-level orchestrator which manages the underlying infrastructures or (2) using a network overlay that federates the network across the IaaS domains. However, combing different types of cloud services together with different underlying network technologies meet challenges such as migration of services, network control, network topology changes and network isolation. One network paradigm that aims to solve these concerns is Software-Defined Networks.

2.1.2 Software-Defined Networks (SDN)

The need for automating the network configuration in cloud networks emerged in a concept of SDN. SDN is defined as a paradigm which separates the control plane from the underlying network data plane. This allows a more efficient control of both network management and network control. Further, centralised control allows fast provisioning and network programmability.

The idea of centralised control and network programmability goes back to the late nineties [21]. However, the term SDN originated from a research article from Standford in 2009 [22]. This article introduced OpenFlow [23] as a new network control plane protocol based on the concept of making flow-based centralised forwarding decisions. Hence, SDN is in the academia often associated with OpenFlow. However, SDN is not only OpenFlow. The industry uses the SDN term

more widely, where the infrastructure vendors claim they comply with SDN as long as they have a centralised controller which distributes network configurations through Application Programmable Interfaces (APIs). For example, Google B4 [24], Cisco ACI [25], Fortinet's FortiManager [26] and VMware NSX [27] are all vendors which use protocols such as OpFlex [28], BGP FlowSpec [29] and NETconf [30] to distribute network configuration following the SDN paradigm [31].

Most SDN protocols, such as OpenFlow, OpFlex and FlowSpec for the Border Gateway Protocol (BGP), only define a framework of how to forward packets on the data plane. They do not handle equipment deployment and network operations. In operator networks, such as in cloud computing networks, OpenFlow does not handle how to set up the basic configurations and how to handle operational aspects such as troubleshooting and lifecycle management. Additionally, the flexibility of most SDN protocols is limited to a lack of standardisation. For every new protocol, the OpenFlow standard must be updated in order to support any new header attributes, and a new software release must be distributed to the switches.

Open Networking Forum (ONF) [32] addressed these challenges in their stratum project where they have introduced a set of next-generation SDN capabilities. Here they defined interfaces such as:

- **P4** - A programming language for switches where the programmer can define a customised pipeline of packet processing and define a customised set of forwarding rules (However, P4 was not defined by the ONF, see section 2.7 - P4).
- **gNMI [33]** - Google Remote Procedure Call (gRPC) Network Management Interface (gNMI) is an interface that uses OpenConfig models for configurations and telemetry
- **gNOI** [34] - Google Remote Procedure Call (gRPC) Node Operator Interface (gNOI) is an interface for operational management and troubleshooting.

In a cloud computing context, SDN is an important contributor in providing dynamic and flexible networks. Especially, the aforementioned next-generation SDN interfaces open up for a more agile deployment of virtual switches. However, when interconnecting multiple clouds with different flavours of SDN, the interconnection standards and the security aspects remain as a challenge. SDN controllers are mainly designed in a vertical manner, primarily for single controller operations. Interconnecting multiple controllers across cloud computing domains introduces a new problem of how to interconnect the SDN controllers horizontally.

In a multi-cloud environment, it is possible to interconnect the controllers in a horizontal east-west approach using a protocol such as BGP [35] or SDNi [36]. A second approach is to use a hierarchical model with abstraction layers by interconnecting the controllers in a vertical manner, such as the Forwarding and Control Element Separation protocol (ForCES) [37]. Here, there are different levels of abstractions and interconnection methods, where the most common method is to use a top-level orchestrator to distribute network policy rules to subordinate SDN controllers. A third approach is to only use one SDN controller and solve the interconnection problem by virtualising the underlying physical network and configure an overlay network. This is commonly known as network virtualisation.

2.1.3 Network Virtualisation

Network virtualisation consists of creating a network within a network. This concept is often represented as an overlay network which consists of one or multiple underlay networks. This is very similar to a Virtual Private Network (VPN), but a virtualised network can also include virtual routers and virtual switches which primarily exists in the overlay network. The concept of overlay networks originated from a need in cloud networks with multi-tenancy and IaaS. In the cloud model, the resources which are provisioned for every tenant are connected to a virtual network for each tenant. The challenge here is that each data centre tenant demand a seamless migration between different physical compute hosts. One example of this is Virtual Machine (VM) migration between off-premises and on-premises resources. These resources often exist in two different physical networks, separated by layer 3 routers. In order to maintain a layer 2 interconnection between two virtual machines during service migration, the virtual machines must be interconnected by a dynamic layer 2 virtual overlay network. From an operative perspective, this is most frequently solved by running a virtual switch on each physical server/compute node, which is interconnected by the use of a tunnelling protocol such as the Stateless Transport Tunnelling Protocol (STT) [38], Virtual eXtensible Local Area Networks (VXLAN) [39] or Generic Network Virtualization Encapsulation (GENEVE) [40]. This network overlay concept is referred to as a Network Virtualisation Overlay (NVO). The standardisations of the different NVO protocols are mainly driven by an IETF workgroup (WG) named NVO3 WG [41].

However, even if the concept of network virtualisation originated in intra-data centre networks, it can also be extended to include multiple data centres which are defined as federated overlay networks. Federating networks means to share resources among multiple independent networks in order to optimise the use of those resources. Accordingly, there are two ways to perceive a federated SDN network. Either by federating the virtual networks as one big NVO with one SDN controller,

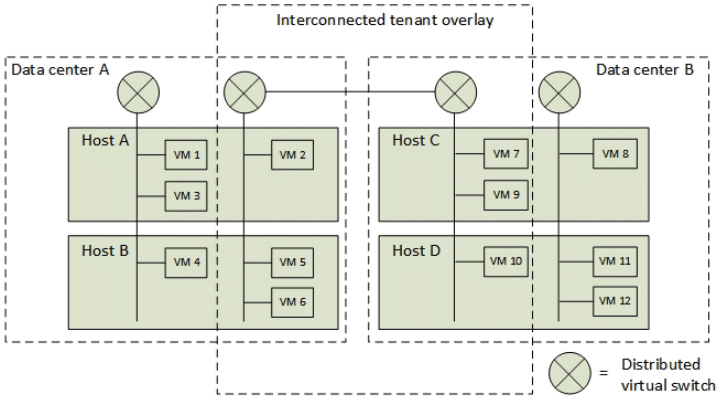


Figure 2.1: Multi-tenancy in virtual networks.

or by using two interconnected NVOs and federating two SDN controllers.

Historically, the main challenge with multi-tenant NVOs was that the virtual switches for each compute node were controlled by the provider's SDN controller, where the tenants did not have any direct control on their virtual networks. This issue has been resolved by letting each tenant create their own virtual switches [42],[43] (Figure: 2.1). However, an NVO between multiple cloud providers with different service models, different hypervisors and different APIs still remains a challenge. Some of these challenges relate to heterogeneous services, management, orchestration addressing schemes, differences in overlay protocols, missing standards and cost models. One organisation that has addressed these challenges is the European Telecommunications Standards Institute (ETSI) through their workgroup of Network Function Virtualisation [44].

2.2 Network Function Virtualisation (NFV)

Network Functions Virtualisation (NFV) is a network architecture concept in which network functions are virtualised. It builds on the idea that traditional network devices and network services can be moved from hardware appliances to virtual software instances which are running on commodity hardware in the cloud. The concept builds on the aforementioned technologies of cloud computing, network virtualisation and SDN. Similarly, NFV relies on the virtualisation technology in order to aggregate physical resources into virtual resources of virtual networks, virtual machines and virtual storage. NFV adopts the on-demand approach from cloud computing and aims to let the end-user request provisioning of network functions. Examples of such network function include switches, routers, firewalls, load balancers, Customer Placed Equipment (CPE) and Router Reflectors (RR).

Running these network functions in centralised data centres transfer the intelligence and the workloads from distributed network appliances into centralised software. This enables a more efficient scaling of resources which similarly to cloud computing saves costs. It also enables the service providers to innovate more rapidly and make them enjoy faster time to market for new services.

The NFV concept originated from a whitepaper which was produced after an SDN conference in Germany in 2012 [45]. The group consisted of representatives from the telecommunication industry, which also was a part of ETSI. ETSI continued this work and designed a high-level architectural framework for NFV, which later has been the origin of the NFV standards. The framework consists of several guideline documents such as targeted use cases, naming conventions, security guidelines, standardisation of components and most importantly, standardisation of API interfaces.

2.2.1 The NFV framework

The NFV framework consists of three main components defined by ETSI [46] (Figure: 2.2).

- **The Virtual Network Functions (VNF)**, are the software instances of the network functions. This corresponds to the virtual end-user service in cloud computing, such as a VM or a container instance. In NFV, a VNF can also be associated with an Element Manager (EM), which handles the configurations of the VNF attributes such as VM naming and IP address configurations.
- **The Network Function Virtualisation Infrastructure (NFVI)**, represents the hardware and software which run the VNFs. This is most frequently associated with the hypervisor domain which controls the compute, network and storage resources. The overlay network (NVO) is a part of the NFVI. This means that the NFVI can span over both multiple physical servers in one data centre or span across multiple data centres.
- **The Management and Orchestration framework (MANO)** is a collection of all the components which are managing and orchestrating the NFVI. In a MANO context, the most important contribution from ETSI, is a set of RESTful API specifications of how the orchestrations system components are intended to make intra- and inter-NFV domain communication. The MANO components are also responsible for resource control and life-cycle management of the VNFs. The life-cycle management is handled by the VNF manager, which includes instantiation, scaling, termination and upgrading of the VNFs. The resource control is managed through the Virtual

Infrastructure Manager (VIM). Most importantly, this includes the allocation of NFVI resources such as network configurations. In comparison to cloud computing, this corresponds to a similar concept of having a template of a group of VMs with a deployable description of the network connections between them. This is also often referred to as blueprinting [47].

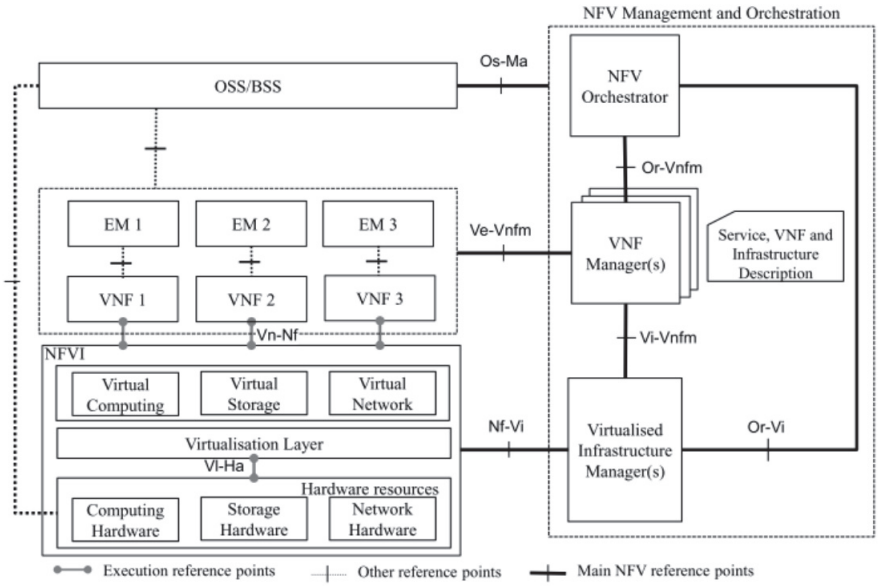


Figure 2.2: The ETSI NFV reference model [44]

From a top-level NFV-domain perspective, it is possible to interconnect the different NFV components in multiple topologies. This applies both to the communication inside of the administrative NFV domain and between administrative domains. ETSI has outlined different architectural options which support the placement of the different NFV functions in different network domains [48]. This also includes how multiple NFV domains can be interconnected. ETSI originally split the NFV orchestrator (NFVO) component into two parts in order to describe the sets of topological reference models. The first two ETSI documents described how the NFVO could act as a Network Service Orchestrator (NSO) or a Resource Orchestrator (RO) in a hierarchical composition of multiple NFV domains. In this model, the NSO represents the top-level orchestrator, while the RO represented the subordinate NFV orchestrators. One example of this is that one top-level NFVO orchestrates multiple ROs.

However, a more recent approach to describe topological NFV models and how to interconnect multiple administrative domains were released by ETSI in January

2018 [49]. This document simplifies the interconnection models and introduces a set of naming conventions inherited from cloud computing. ETSI identified that multi-site and multi-tenant orchestration were the key priorities in describing NFV interconnection models for ISPs. Hence, ETSI simplified the models. They defined NFVI as a Service (NFVIaaS) and Network Services (NS) in order to describe the interconnection models. In order to describe them in more details, they introduced two uses cases of interconnecting NFV domains.

- **The NFVIaas use-case** is an example of a service provider (the consumer) who runs the VNFs inside an NFVI operated by a different service provider (the producer). Here, the NFVIaaS consumer controls the VNF applications, but they do not control the underlying infrastructure such as the VIM and the NFVI. This implies that the provider is using a remote VIM and a remote NFVI to instantiate their services, but the management of the VNFs is performed in their own administrative domain.

- **The NS use-case** gives an example of a service provider who runs almost all NFV functions in a non-administrative remote data centre. Here the remote data centre only provides access to an NFV orchestrator interface. This implies that the consumer does not handle life-cycle management of the VNF and the consumer must operate multiple catalogues of VNF services.

Interconnecting two or more NFV administrative domains introduce new security threats, where ETSI has listed ten security objectives for this [48]. Their security research objectives are based on a set assumption, where the primary adversary model is that an attacker can intercept and modify data between two NFV domains. This questions the security features of the data plane protocol and the access protection of the API interfaces to the NFV domains.

2.2.2 Service Function Chaining

Service Function Chaining (SFC) is a concept of making a virtual chain of virtual network functions in order to steer the data traffic through these functions hop by hop. This capability can be used by ISPs to provide a set of virtual network functions to end-users or network operators such as firewalls and intrusion detection systems. Setting up an SFC enables operators to create Virtual Links (VL) between every virtual network service and implicitly apply all these consecutive network functions to the network packets. In total, this group of functions is called a Network Service (NS). The SFC can be perceived as the core element in NFV, which is the enabler of the functionality of letting an end-user move his physical residential network equipment from his home and into the cloud.

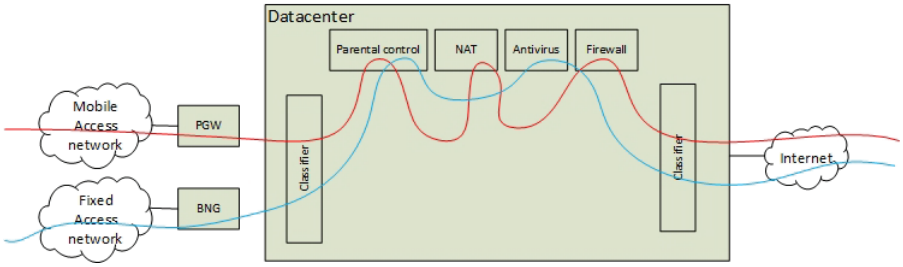


Figure 2.3: A Service Function Chain (SFC)

Using SDN in the NFV framework can enhance the traffic steering capability between the VNFs in order to make them dynamically adapt to service requirements and network changes. An important differentiation when defining the term SFC, is to separate the high-level specification of an SFC and the low-level SFC. The high-level specification of an SFC, the VNF Forwarding Graph (VNFFG), is defined in the OSS, while the low-level SFC is the actual SFC deployed into the network. This separation of abstract and concrete SFC definitions is made because the availability of the resources and the physical locations of the VNFs can vary. In this thesis, it is defined that; the VNFFG is an abstract representation of how the VNFs are intended to be interconnected, while the SFC is the actual network configuration and a result of a rendered VNFFG. However, SFC is not a network protocol header. In this thesis it is defined as a generalised term for any SFC packet header, such as the Network Service Headers (NSH). The SFC, the low-level configuration of the network service chain, is the main focus in this research.

According to the SFC specification [50] by IETF, there are two important attributes to an SFC. (1) The VNF has two virtual interfaces, one for incoming traffic and one for outgoing traffic. (2) An SFC can have a different return path than the forwarding path. However, the concept of an SFC has been developed in many directions by different industry groups, where the approach to an SFC and the naming conventions are slightly different.

The ONF has focused on using Policy-based Routing (PBR) for enabling the SFC. ETSI, together with Cisco and IETF, have suggested using Network Service Headers, but the IETF also focus on SFC in MPLS. However, it can be questioned if all these different technologies are capable of advanced traffic steering between data centres. Specifically, this research questions if all these technologies are compliant with the aforementioned attributes of the SFC specification (Section: 2.3). However, what they all have in common, is that they are extracting data out of the packet headers in order to steer the SFC packets.

These SFC instructions extracted from the data packets is essential for all the nodes participating in steering the data packet through the SFC. These SFC instructions typically only reside within the boundaries of the SFC domain. The SFC specification defines that a Classifier Function (CF) can add these SFC instructions to a data packet (or alternatively get the flow specification of the packet path). This concept of packet classification represents an important difference in multi-domain network interconnections. In overlay networks across multiple administrative domains, these addressing scheme and network structure are global within the overlay. However, in heterogeneous networks which are not using a shared overlay network, these SFC instructions were historically intended to be removed from the packet when exiting one domain. Consequently, this results in packet reclassification when the packets enter a new administrative domain. The reclassification was needed due to different addressing schemes, different SFC technologies and different control plane technologies. However, it is possible to map the packet attributes of two different SFC protocols or administrative domains together (Figure: 2.4). This is commonly known as network stitching.

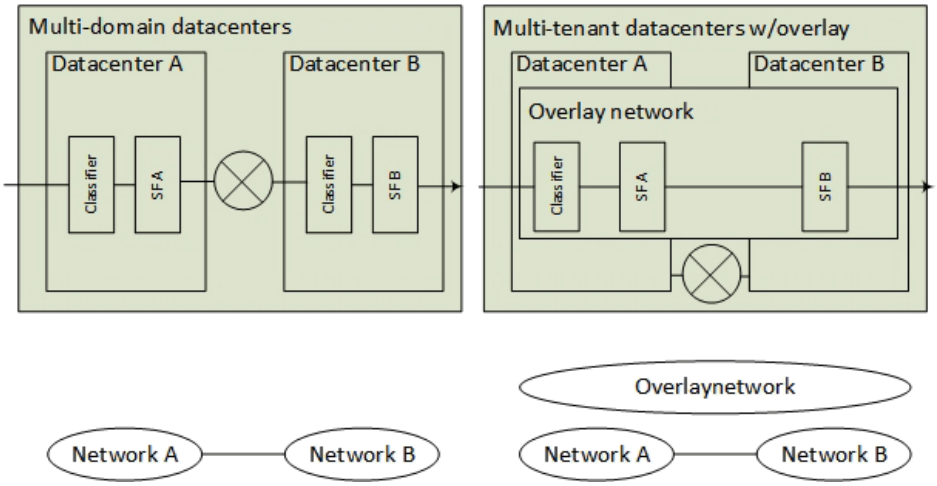


Figure 2.4: Classification of an SFC

In this thesis, these two concepts are defined as overlay networks and multi-domain network orchestration (MdO).

A research challenge with an SFC has previously been that the traffic steering mechanism has to handle state information for each hop in an SFC. This is a network packet forwarding mechanism which is mainly driven by the IETF.

The state problem is exemplified in a use case where a packet traverses the same router multiple times (such as router B and C in Figure 2.5). These intermediate

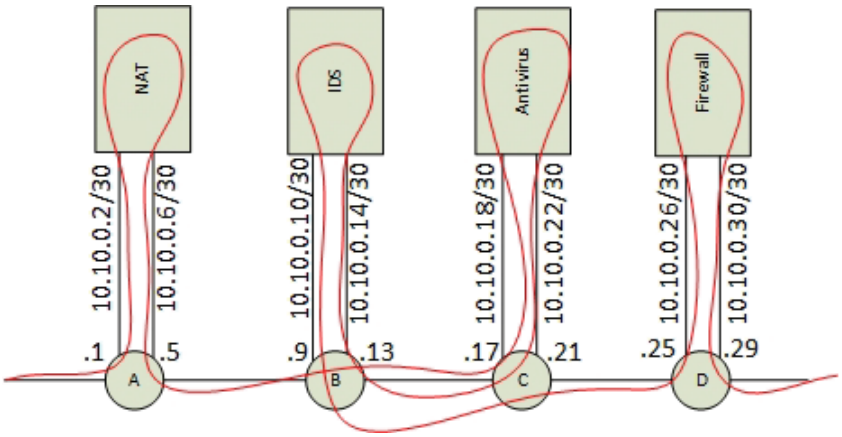


Figure 2.5: State problem in an SFC

routers have to keep track of how many times the packet has traversed the routers in order to make the correct routing decision. This is a typical SFC scenario. Figure 2.5 shows how router C receives the same IP packet on the same interface twice when the SFC path is NAT -> Antivirus -> IDS -> Firewall. The first time router C receives the packet from router B, its' destination is the antivirus VNF, while the second time it receives the same packet from router B, its' destination is router D.

Source-based routing does not solve the problem because the source and the destination in the IP packet do not change along the path (except for the NAT VNF hop). Neither do source routing based on MAC addresses or interfaces solve the problem. In a plain layer 2 network, where the routers are replaced by switches, it is possible to use the source and destination MAC and IP addresses to make the correct routing decision. However, because the SFC standard also enables to let a VNF being used multiple times in one SFC, the MAC address is neither possible to use.

Interconnecting multiple data centres with intermediate routers and shared VNFs calls for the same problem when an IP packet traverses back and forth between two data centre sites (Figure: 2.6). Correspondingly, there is a need for a packet header which keeps track of the SFC hops. There are primarily three solutions to this problem. One solution to keep track of the number hops, is to create a full mesh of overlay tunnels between every VNF. This also implies the establishment of multiple tunnels between pairs of VNFs. A second method is to introduce a packet header which identifies the SFC and counts every SFC hop (i.e. NSH, MPLS). A third method is to encode every hop into the data packet, namely Segment Routing (SR).

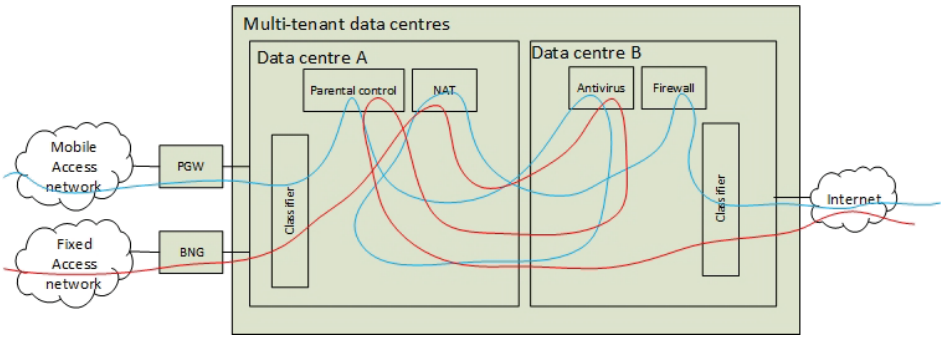


Figure 2.6: SFC state problem between multiple data centres

Segment Routing

The three aforementioned methods are based on two routing principles. (1) Letting each individual router make forwarding decisions based on attributes from the packet headers and the tunnel interfaces and (2) Letting the packet sender encode the routing path for every hop into the data packet. In an SFC context, this translates to stateful and stateless SFCs [51]. The stateful option is to track the SFC state by determining the next hop based on a packet header attributes (i.e. the TTL field, a tunnel or a combination of layer 2 and layer 3 headers). On the other hand, segment routing is a stateless approach for enabling an SFC. Segment routing aims to explicitly indicate the forwarding path for the packets at the ingress node by inserting an ordered list of instructions directly into the packet. These instructions are called segments and typically contains an address identifier for each router hop.

The Source Packet Routing In NetworkinG (SPRING) working group in the IETF has defined Segment Routing (SR) for MPLS (MPLS-SR [52]) and IPv6 (SRv6 [53]). Both segment routing options are available as the underlying network protocol for steering the SFC traffic. The main problem is that MPLS-SR is not widely adopted in data centre infrastructures and that SRv6 does not support IPv4 traffic from the end-user. There have been attempts to make NFV SFCs by SRv6 [54], but IPv6-IPv4 translations are challenging for dynamic NFV infrastructures.

Stateless SFCs are very attractive since it does not require much resources from the intermediate routers. Neither does it require complicated control plane integration between interconnected service providers. However, in general, SR is not very suitable for SFC. The fixed set of SFC hops implemented in a packet is not very dynamic and secure. SR, in general, also makes re-directions inside VNF complicated. This is challenging because it requires a lot of resources to recalculate the list of segments for every packet. Most importantly, it is also challenging to

combine packet encryption with segment routing, because packet encryption also can encrypt the list of segments (i.e. SRv6 [53]).

Hence, this research is focusing the work on stateful SFC routing, which also includes an SFC packet header and a corresponding component capable of forwarding such packets. This component is named the Service Function Forwarder (SFF). The SFC header NSH and the SFF are discussed in the next two sections.

Network Service Headers

Network Service Headers (NSH) is a protocol specification of the abstract concept of the SFC header. It provides a transport and topology independent service forwarding framework customised for overlay networks. The NSH protocol enables the VNFs to exchange metadata across each other (optional header fields) and it allows the intermediate forwarders (SFFs) and VNFs to change the SFC path (classify or re-classify flows). It also aims to provide end-to-end service path visibility, but it relies on the integration of the overlay networks. The packet header contains two primary identifiers, a Service Path ID (SPI), which identifies the SFC, and a Service Index (SI) value, which is intended to decrement for every SFC hop.

Service Function Forwarders

This SFC overlay introduces a new problem in the NFV network. In the traditional OSI reference model, there are 7 layers which each represent distinct layers of forwarding responsibilities. Traditionally, the main layers of network forwarding have been based on layer 2-4, which traditionally have been handled by switches (layer 2) and routers (layer 3). The concept of this new SFC header did not fit into this model, and neither was the corresponding packet forwarder role defined (switch or router). Hence, in the SFC specification, IETF introduced a new component, namely a Service Function Forwarder (SFF). The role of this component is to make forwarding decisions based on the destination of the service functions. However, due to different sets of SFC forwarding mechanisms, the specification enables the SFF role to be placed on multiple OSI layers, more specifically between OSI layers. Most importantly, this new network component is capable of making forwarding decisions based on the SFC. One perception of the role of the SFF, is that the SFF makes forwarding decisions based on a new packet header, the SFC header. However, SDN technologies such as OpenFlow enables these forwarding decisions to be based on a combination of packet attributes of multiple packet headers. In this research, the SFC header is referred to as a generalised new packet header and the SFF as the network component capable of making forwarding decisions based on this packet header.

Most commonly, the SFF represents a virtual switch placed in each compute node.

For example, in the Open NFV platform (OPNFV) [55], the SFF is an Open Virtualised Switch (OVS) with OpenFlow support [56] which is placed on each Compute Node. The SFF forwards SFC packet to other SFFs or to the VNF connected to the Compute Node. In most NFVIs, the SFFs are usually interconnected by tunnels between every SFF (i.e. IPsec [20], STT [38], VXLAN-GPE [39], GENEVE [40]). In the SFC specification, these tunnels are referred to as transports tunnels. Note: In this context, the phrase "transport" is not directly associated with the transport layer in the OSI model. Here, this represents a tunnel which transports the SFC packets between the SFFs.

When interconnecting multiple data centres, the topology of these transport tunnels represent different interconnection methodologies. They can for example span over multiple data centre sites in a common network or they exist in two different administrative domains interconnected by an intermediate edge router. Running an SFC between two data centres using different SFC technologies is therefore challenging. In order to preserve the SFC state of the packet, the data packet must either maintain the SFC header while it crosses the data centre borders, or the packet must be re-classified. However, when encrypting an SFC packet, reclassification and general SFC routing meet a new challenge.

2.3 The SFC encryption challenge

In NFV, the very fundamental concept is that a network packet, arriving from for example an end-user, has to be processed by intermediate middleboxes (VNFs). In many cases, the original data-content in a packet from the end-user is not changed while the packet traverses from one service to another. In order for the packet to be processed by the VNFs, the packet should preferable be non-encrypted (or by using SSL inspection on VNF ingress)[1]. Hence, the research problem is based on the assumption that end-user data packets are non-encrypted in the SFC. This makes the packets vulnerable for eavesdropping, manipulation and redirection inside the NFV data plane domain.

The approach to solve the problem in this research, is by introducing packet encryption and packet integrity checks inside the SFC domain. This is achieved by enabling packet encryption hop by hop, which is collectively perceived as end-to-end encrypted.

The second research objective is access control, which implies to isolate the VNFs from having access to the end-user data packets. This research aims to solve this by encrypting specific data flows across specific VNF hops. Consequently, the VNFs, which do not have access to the data, are bypassed. Regardless of the SFC

[1] Standard SSL inspection resolves the issue of encrypted SSL packets [57, 58].

routing rules, the access list controls where the encryption keys are distributed and what VNF which potentially can access the data. In the research use case, it is defined that the encrypted data is routed through the VNFs.

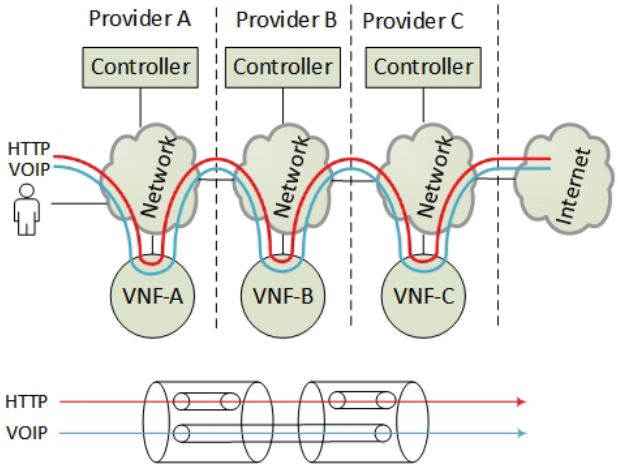


Figure 2.7: Hop by hop encryption

Hence, this research aims to ensure that the data traffic is secured from eavesdropping from all unauthorised parties. Figure 2.7 exemplifies this concept: VNF A can access the HTTP traffic and not Voice over IP (VOIP) traffic, while VNF B can access HTTP traffic, but not VOIP. VNF C has access to both HTTP and VOIP. Currently, this functionality is not supported in the SFC standard from the IETF [50].

It is mentioned previously that combining packet encryption with segment routing or SFC headers is challenging. This is because packet encryption can hide packet data which is needed for the intermediate switches and routers to forward the packet. However, this highly depends on the technology which is being used to control the packet forwarding.

This problem is demonstrated by showing 5 examples of packet structures (Figure: 2.8). Note: For simplifying reasons, TCP/UDP is included in the IP header block in the Figure.

1. Figure 2.8-1 gives an example of a plain IP packet which is encrypted and contains no SFC header. If the packet is encrypted by IPsec, the SFFs are not capable of steering the packets based on TCP/UDP port. It is neither capable of looking up any SFC state in the packet.

2. Figure 2.8-2 shows an example of an IP packet containing an SFC header.

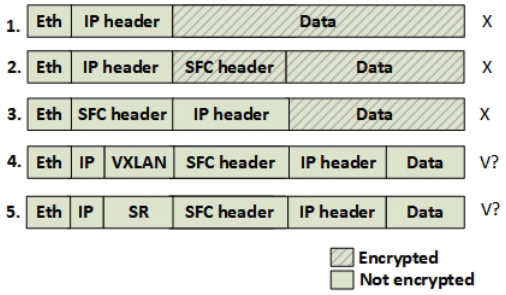


Figure 2.8: Packet encryption strategies

Encrypting the IP packet then also results in encrypting the SFC header and makes it similar to packet example 1.

3. Figure 2.8-3 demonstrates how an SFC header can be placed outside the IP header. Here the packet can be routed based on an SFC header. However, there are currently no standards of mapping Ethernet MAC addresses to SFC addresses (such as ARP). Also, when the packet traverses a layer 3 router, the SFC header is potentially lost.

4. Figure 2.8-4 shows an example of an outer IP tunnel that encapsulates both the SFC header and the IP header. This simplified packet structure shows how the outer IP tunnel is transporting the packets between the SFFs. Here, the outer IP tunnel is only valid per SFF hop. In such a packet structure, it is possible to apply encryption and isolation measures. Correspondingly, this research aimed to extend this packet structure by applying encryption and isolation capabilities.

5. Figure 2.8-5 gives a similar example to example 4, but here with segment routing being used as a tunnelling mechanism between the SFFs. This principle is based on a recent draft from, IETF (June 2019) where they proposed to combine the concepts of stateful and stateless SFCs by introducing NSH-based SFC with SR-based transport tunnel and SR-based SFC with Integrated NSH Service Plane [59]. This is a new alternative solution to the problem which have not investigated in this research.

The two last examples demonstrate packet structures which makes it possible to apply encryption and integrity checks. It is possible to apply this to both the inner and the outer IP packet. Applying it to the outer tunnel can ensure confidentiality between every SFF, while encrypting the inner packet hop-by-hop can ensure

confidentiality for the traffic inside an SFC. However, it does not solve the access control problem within an SFC. Using different encryption keys on different flows within an SFC requires that the different flows also are identifiable in a forwarding perspective. This is because different encrypted packets must be handled by different encrypting functions. In this research, this is referred to as the flow identification problem of encrypted packets (Chapter: 8). Consequently, this information must be extracted from an attribute in a packet header. This research aimed to solve this by adding this information into the SFC header.

In summary, this section has shown the need for stateful SFCs, which includes attributes for identifying encrypted flows for access control. In the context of interconnecting multiple data centres, this raises two additional problems.

First, these SFC identifiers are located in the data packets. Correspondingly, the packet header identifiers have to be coordinated across all SFFs and to any relevant encryption function. There must either be a coordinated mapping (stitching) of the identifiers between different domains or they must exist in one domain (overlay).

Second, it calls for synchronising the network configuration on multiple levels. This includes both the setup of the underlying SFF tunnels, the setup of encryption parameters and the setup of the SFCs. This coordination of network configuration can either be (1) performed on a top-level orchestration level or (2) by an interconnected control plane. Both methods question the need for a vertical provisioning of network resources.

These problems reflect research question one, two and three, where there is a need to identify the interconnection method, the requirements and a solution to this problem. Generally, the need for the distribution of advanced network configuration calls for centralised control and programmability, which reflects SDN capabilities.

2.4 Encryption as a service function

There is currently no specification of how to enable isolation and encryption of SFC traffic in NFV. The SFC specification [60] only states that the encryption can be ensured by the NFV transport protocol. In this context, the transport protocol layer is not the OSI model layer 4, but it is the outer transport layer that transports a data-packet between two VNFs. This layer often corresponds to the Virtual Link channel between two VNFs. The transport layer can enable encryption, but the consequences of applying encryption depend on the transport protocols' encryption capability. In most cases, applying encryption to an IP packet results in a lack of readable headers, which prevents routers from routing packets correctly through an SFC.

2.5 SDN in NFV

SFC orchestration is a key problem in NFV [61]. The main problem is the VNF deployment, where the physical resources, such as VMs and multiple virtual networks, must be integrated both vertically and horizontally. One aspect of this, is the aforementioned problem of the provisioning of multiple network layers. A second aspect, is that the network and SFC path is dependent on the attributes of VMs and vice versa. For example, a full network link should result in a re-provisioning of a VNF in another data centre, while a failure in a VNF application should result in a reconfiguration of the SFC. This also calls for integrations across different virtualisation domains, such as compute, storage and networking. From a networking perspective, Traffic Engineering (TE) is consequently dependent on more attributes than the traditional bandwidth and latency parameters. It also has to take data centre attributes into account, such as on-demand resource scaling and failure recovery. Hence, centralised network control is essential in order to manage all the resources.

The previous section clearly stated the need for distributed network configurations on multiple layers. Correspondingly, it is possible to apply the SDN paradigm to different sets of network resources in different network domains in NFV. SDN can be used to control the resources inside the overlay network, such as the VNF, or it can control the resources in the underlying network, such as the SFFs or the physical switches in the NFVI. Also, it is possible to apply multiple SDN domains within one NFVI domains.

These demands correlate with the ETSI specifications of SDN and NFV [48]. ETSI enables the SDN controller to be placed in the VNF, the NFVI, the VIM or in the NFVO (Figure: 2.9). However, it also questions if there is a need for multiple SDN controllers which are interconnected vertically.

Originally, NFV in general, introduced a need for combing data centre resources with network configuration. Hence, NFV is perceived as an attempt to control both. This need introduced the abstraction layers of NFV, which visualise the connection between the compute domain and network domain.

In this context, there have been many attempts to achieve this simplified architecture (i.e. T-NOVA [63], UNIFY [64]). The need for consolidation came from different naming standards and technologies in different organisations. The UNIFY project consolidated the work from ETSI (NFV) and ONF (SDN), which derived a model with three abstractions layers. The service layer, the orchestration layer and the infrastructure layer.

The infrastructure layer manages all data centre infrastructure and network re-

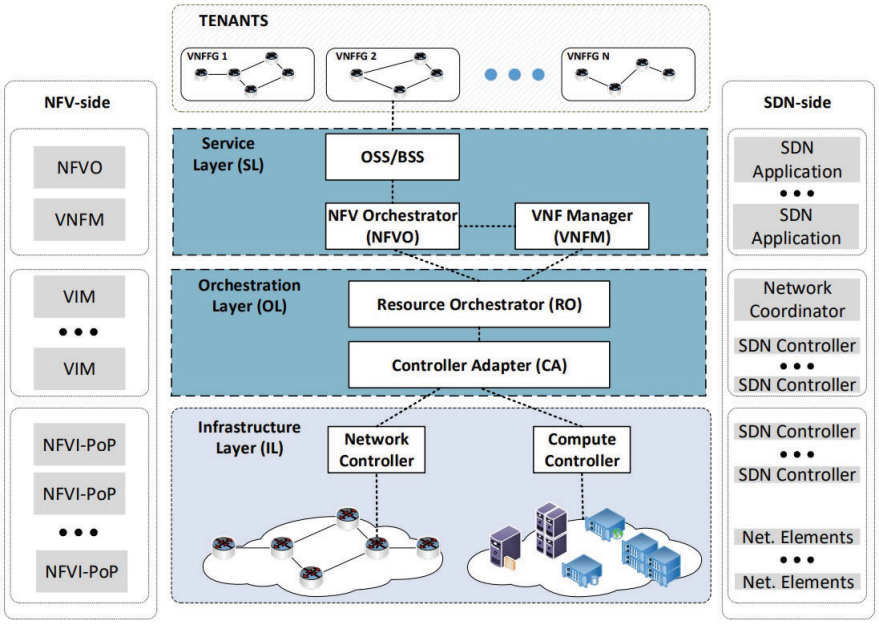


Figure 2.9: NFV and SDN relationship [62]

sources such as OpenFlow switches and OpenStack enabled data centres. The orchestration layer handles the orchestration of the data centre resources and the integration of the virtual network resources. The service layer compromises the orchestration related to Business Support Systems (BSS) and Operation Support Systems (OSS). Hence, it can also orchestrate different orchestration system. Most importantly it executes the NFVO and the VNFM functions in order to instantiate the VNFs and distribute the network configuration.

In summary, there are interconnection models coming from the NFV-side and from the SDN-side. Together, the different interconnection points are open to different strategies.

An important premise for these studies is that the answers to the research questions aim to have an impact on the real world. This implies that a solution to the encryption and isolation problem in SFCs should be aligned with the latest research and trends in the market. Choosing an interconnection strategy should therefore not only be based on research contributions, but it should also be relevant for the market.

2.6 Industry impact on NFV

Interconnection methods of virtualised networks have been continuously studied since the appearance of Software-Defined Data Centres (SDDC). This includes a wide range of research domains including cloud computing, SDN, IoT and NFV. Among these research domains, the research challenges are similar and overlapping, but they focus on different application domains. These research silos are perceived as a challenge for NFV, but also as a potential for having synergy effects across these research domains. An additional factor which brings an additional dimension to these research silos, is that the related work of NFV is not only originating from the academia but also from private organisations and the industry. Hence, with respect to the research question (RQ-1) of identifying NFV interconnection methods, the related work is not only driven by academic contributions, but it is also driven by trends in the industry. In order to let this academic research have any impact on the society, it should be aligned with these trends.

ETSI aims to lead, consolidate and coordinate the work of NFV standardisation [7]. Different research and technology silos have tended to work independently or in parallel with different focus areas. However, since 2012, ETSI has released over 100 publications (Figure: 2.10) aiming to have one common NFV standard. They have published consolidation, capability, requirement, architecture and specification papers and ETSI is now focusing on consolidation and optimisation. However, the publications from ETSI mostly include APIs for the NFV components and not network protocols or virtualisation techniques. This reflects the focus areas of ETSI, which primarily aims to interconnect both components and data centres by APIs and not by distributed network protocols. ETSI aims to describe the network services in an abstract language (i.e. NETCONF/YANG [30], TOSCA [65]) and to distribute this configuration between the NFV components located within or between data centres.

While ETSI focuses on interconnections by APIs, IETF and IRTF have a more network-oriented approach to NFV interconnections by focusing on standardising network protocols on the data plane. Their primary focus in an NFV context has been an SFC architecture with an interface between the network and the virtualised services (I2NSF [66]). Similarly to ETSI, standardisation organisation such as MEF [67] has envisioned the service perspective, while ONF similar to IETF focuses on network forwarding and network control. On the other hand, standardisation organisations closer to the physical network layer, such as optics [68] or mobile networks [69], have a very specialised focus on how to interconnect the data centres. For example, 5GPP [69] focus on Software-Defined Radio (SDR) access networks and how to interconnect them across multiple service providers.

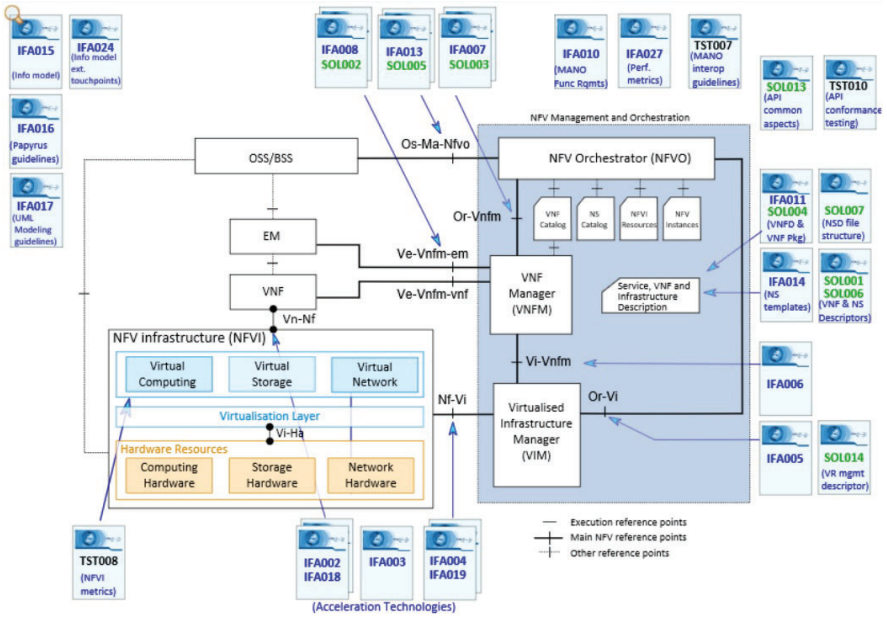


Figure 2.10: ETSI specifications [6]

They also envision new service concepts, such as network slicing [70], but their architectural concepts of interconnecting different domains have historically not been fully aligned with ETSI. ETSI focus on openness of multiple virtualisation techniques through overlay networks and standardising APIs. The 5G community is focusing on a uniform communication model between the operators.

A simplification of the main difference is that ETSI aims to use overlay networks when interconnecting to other operators, while the 5G community aims to use a common control plane and a common networking standard (MdO). These methods correlate with the SFC interconnection methods mentioned in Section 2.2. These interconnection methods are perceived as two different strategies and can be defined as stratification strategies (Figure: 2.11).

With respect to Service Orchestration (SO) and Resource Orchestration (SO), the different responsibilities of these orchestration layers are highly dependent on the top-level stratification strategy. Resource orchestration of a virtual network inside an interconnected single overlay network is naturally easier than adapting and converting between different network standards in different domains. However, an additional network overlay introduces network overhead and operational challenges. In addition to having different stratification strategies, the different standardisation organisations also work with different use cases (with different terminologies)

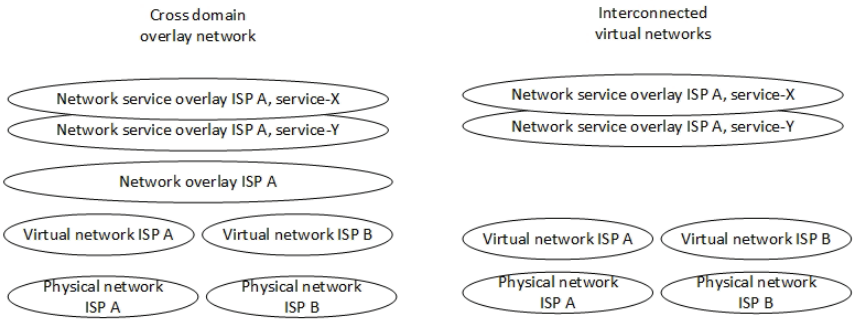


Figure 2.11: Stratification strategies

which reflects their suggestions of interconnection points and interfaces between the NFV service providers.

As stated previously, ETSI primarily works with federation across NFVOs, while MEF works with orchestrating VNF services [67]. Additionally, the Telecommunication Management Forum (TMF) [71] work with operational aspects and naming standards (TMF 633, 640, 641, 645, 653, 656, 677) [71], while the Next Generation Mobile Networks (NGMN) [72] and the 5GPPP [73] work with 5G use cases for NFV [74].

However, from June 2019, ETSI and 3GPP SA5 [75] are aiming to collaborate and incorporate the principles from 5G into the NFV standard. This also includes network slicing.

Additionally, the open-source community has been contributing to making NFV standards. Their agile development strategy of "code first", has resulted in a set of successfully NFV applications that uses different API standards.

ETSI aims to consolidate the actors in all these domains, where they aim to collaborate with both operators, vendors, standardisation organisation, the open-source community and the academia.

When ETSI joined forces with the open-source community, the operators also gained more focus in adopting the ETSI architecture. Hence, the open-source community have been important contributors in connecting the real world and the operators to adopt NFV.

Open Source Management and Orchestration (OSM) [76] is now an ETSI-hosted project set to develop an open-source orchestrator for NFV. The NFV orchestrator ONAP (formerly OPEN-O) [77] is perceived as the competitor to OSM. The competition and their different standards did split both operators and vendors.

European operators and vendors chose OSM, while Asian and North American vendors chose ONAP. In April 2019, the organisations join forces and are now working together towards a common NFV orchestration model. During the past years, there have been many similar consolidations of standardisation organisations and the open-source community. MEF is collaborating with Opendaylight [78], while Opendaylight is chosen as the main SDN controller in Openstack [79]. Openstack has for many years been the leading open-source framework for cloud computing and they were also the origin to the leading open-source platform for NFV (OPNFV) [55]. Similarly, the NFVO system CORD [80] grew out of the SDN controller ONOS [81].

This stream of collaborations reflects how the different NFV research domains are merging. However, even if OSM and ONAP are now collaborating in the NFVO domain, the NFVI domain is more diverse. Historically, vendors had different interpretations of how to implement the ETSI specifications. ETSI, which aligned with OPNFV, defined the OPNFV platform as the NFVI standard for interoperability tests. According to the interoperability tests from ETSI, most vendors are now compatible [82]. However, research challenges still remain related to the interconnections of NFVI infrastructures, such as isolating virtual services [83], which is also the research objective in this thesis.

In summary, ETSI tends to lead the work of interconnecting NFVI domains, and their work on standardising the APIs between different operators seems successful. However, their work on interconnected network domains is open for multiple models. The platform which tends to be most successful and also most used for interconnection test is the OPNNFV platforms. For interconnecting these different NFVIs, the flexibility and openness in the platform is acknowledged. Hence, this research aims to be aligned with both ETSI and the OPNFV platform.

2.7 P4

The different approaches to segment routing and SFCs have resulted in multiple standardisation attempts (Section: 2.6) of supporting the underlying network infrastructure in the forwarding SFC packet. Also, NFV research, in general, has resulted in an increasing number of standards, header extensions and API interfaces. This introduces a new problem where some standards are developing so fast that a new version of a standard exists before the last version is implemented. The background for this problem is that every new protocol typically goes through the IETF board, which in most cases is a slow process. Additionally, chip designers usually spend time to implement the new protocols into hardware. However, even if the process was performed much faster, operators cannot afford to change their equipment very often.

Hence, a response to this problem is the Programming Protocol-Independent Packet Processors (P4). This is a programming language which allows switch operators to program the packet forwarding plane and implicitly customise the packet forwarding rules in the switch. It originally appeared in a SIGCOMM CCR paper in 2014 [84], while it is now mainly driven by the organisation Barefoot [85].

The concept of P4 builds on the virtualisation principles from the application programming domain, which targets an independent abstraction layer of application interfaces, which conform a programming framework similar to JAVA and .NET. This implies that a P4 program can run on any switch and any operating system. It builds on a Protocol-Independent Switch Architecture (PISA), which ensures that the switch program can support any packet header format specified in the program code. From a control-plane perspective, P4 is control plane independent and allows any control plane agent to dynamically specifically customised flow rules (Figure: 2.12). As mentioned in Section 2.1.2, the programming framework is included in the next-generation SDN framework, but it is currently also widely available on both virtual or physical switches [86].

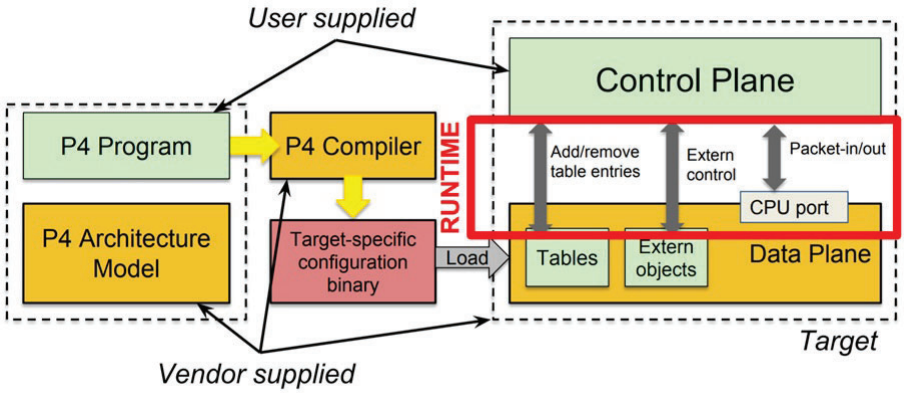


Figure 2.12: P4 architecture from p4.org

This research aims to utilise the P4 platform in order to support customised packet forwarding in a virtualised overlay network. High-level overlay networks can span over multiple data centres where an operator can deploy virtual SFFs which supports P4. Hence, this technology is an enabler for advanced packet forwarding including access control and SFF forwarding of encrypted packets.

2.8 Summary

This section has shown how Network Function Virtualisation has taken concepts from cloud computing, virtualised networks and SDN and evolved into a virtual-

isation technology which handles both compute and network resources in multi-domain topologies. This history has resulted in different perceptions of how the underlying resources are abstracted. In a networking perspective, this has resulted in multiple protocol standards and interconnections methods. This is also reflected by overlapping research areas and application domains between NFV, SDN and cloud computing. Additionally, the NFV innovations are driven from the academia, standardisation organisation, the industry and the open-source community. Hence, there is a need to consolidate the technologies and evaluate the capabilities and security implications when interconnecting NFV domains.

Simplification of networking, virtualised network services and principles of overlay networks with micro-segmentation have resulted in a lack of isolation and encryption capabilities in the NFV. The principle of having an intermediate VNF implies that the network traffic cannot be encrypted end-to-end. This is because the intermediate VNFs must have the capability to process the data packets. This questions if NFV, in general, is ready for the general public if this security problem is not solved. Hence, this research aims to solve this problem by using principles from micro-segmentation and introduce isolation and encryption per SFC hop. In order to achieve isolation and encryption, the solution proposal in this research is to instantiate multiple tunnels per user, per data centre, per SFC hop and per VNF, where the network configuration and the encryption-key agreements is automated.

In order to emphasise the aforementioned research objectives (Chapter: 1.3), this research aims to solve this problem by consolidating the different interconnection methods and technologies, define the requirements for a solution, create a solution, implement it and verify the solution by running a proof of concept experiment. This research aims to utilise a new stateful SFC header together with the P4 technology when experimenting on this research objective.

Chapter 3

Related Work

This chapter presents an overview of the related work which concerns this research. The chapter is organised into four parts, which reflects the first four research questions. Accordingly, at first, the related work concerning research question 1 is presented, where it is discussed about interconnected NFV domains and their capability to establish security trust models and exchanging security parameters. Second, this chapter continues by presenting the state of the art of the current interconnection methods of the SFC data plane. These data plane protocols set the foundation of how the requirements in research question 2 are extracted. Third, this chapter presents the most significant research contributions related to the architecture of the network control planes for SFC. Finally, the related work of relevant SFC security applications are presented in alignment with research question 4. Research question 5 encompass the related work from the aforementioned thematics.

3.1 API based interconnection of NFV domains (RQ-1)

Integrating NFV into an operator network is a challenging task. It concerns not only the vertical and horizontal network integration, but it also sets requirements to the standardisation of virtualised network services. Rosa et al. [87] showed a typical use case and research challenge, which involves most ISPs when they are considering providing NFV to their end-users. The use cases consist of having multiple network domains and multiple data centres, which all must cooperate in automating network service delivery. Vaishnavi et al. [88] and Naudts et al. [89] show that the deployment of multi-domain NFV requires an abstraction of the underlying resources. Together with Rothos et al. [90], they state that it is the configuration parameters, the performance metrics and the fault management of the underlying resources which define the top-level interconnection methods. Hence, the first research question reflects this topic, where it is aimed to survey

the end-to-end security models, the different interconnection methods and their dependencies.

According to the ETSI specification [91], NFV orchestration relies on the fact that an abstract high-level VNFFG always must be defined in one OSS domain, while it is further distributed to subordinate systems. Hence, there is always one top-level OSS, often an orchestrator, which initiates the provisioning, but the method it uses to communicate to subordinate systems depends on API standards in these systems. If all subordinate system used one standard and one network technology, the top-level orchestration could be simple. However, multi-domain network infrastructures can contain different network protocols and control interfaces. A design goal for NFV in order to overcome any of these interoperability issues is therefore to have a standardisation and openness of the related interfaces. These interfaces of the NFV orchestrator aims to be abstract definitions, which can be sent to the subordinate systems, where the orchestrators automatically can provision all services and coordinate themselves with the other orchestrators. Neves et al. [92] refer to this as self-organising networks (SON).

Currently, the models are not SON. They reflect a top-down approach, where top-level abstract VNFFG specifications, aim to result in an SFC network configuration on the data plane. This is reflected through the modelling work from ETSI and their attempts to specify NFV service orchestration [93]. ETSI has also modelled a trust framework between different components [94]. However, there is no security model which models the vertical security dependencies between each network layer. Neither is any model found of how orchestration security affects both control plane and data plane confidentiality and integrity. However, in 2019, Rebello et al. [95] explain how the integrity between the NFV orchestration components can be secured in a multi-domain environment. By the use of Block-chain, they achieved both vertical and horizontal trust between the NFV orchestration components.

This lack of a security model and component autonomousity is also reflected in SFC orchestrations. Rotsos et al. [90] and Xavier et al. [96] presents a study where they surveyed standardisation efforts towards enabling network service orchestration from an operator perspective. The study confirms previous studies [97], and shows that there has been a consolidation of orchestration standards. It is shown that a full automation of the subordinate networks and general orchestration are challenging.

A similar study was also conducted by Munoz et al. [98], where they also raised the challenge of having multiple standards. They identified that the underlying reason is that there are many technologies which can interconnect NFV services across multiple service providers. The research shows that, due to a wide set of compon-

ents and abstraction layers, the interconnection of interfaces between the provider domains can be achieved in many ways [98]. This reflects the specifications from ETSI and IETF, where they are open for multiple designs in NFV orchestration and how the components can be distributed. Correspondingly, the main challenge relies not upon the standardisation efforts from ETSI. It relies upon service function orchestration, which relates to (1) controlling the SFCs in the network and (2) managing the VNFs.

First, regarding the SFCs in the network, Hantouti et al. [99] points out that SFC service orchestration yet has many research challenges and that the academia has high focus on elements such as SFC path selection [100, 101, 102], SFC decomposition [103, 104, 105, 104], SFC routing [106, 107, 108], SF placement [109, 110, 111, 112, 113, 114, 115, 116, 117], general security [118, 119, 120, 121, 122, 123] and optimization approaches [124].

Second, regarding the VNF management, Rotsos et al. [90] state that one of the key challenges towards a unified VNF framework is the lack of standardisation of VNF Managers (VNFM). On the other hand, in 2018, Dutta et al. [125] show how the academia, ETSI and MEF have contributed to a common standardisation of interfaces, where they currently also runs periodic interoperability test. However, it can be claimed that the functionality of the interfaces is currently limited to a set of a few basic operations. This is confirmed by Dutta et al. [125], which also shows that dynamic resource management, dynamic allocation, scaling and multi-vendor interoperability still are key challenges. More specifically, it is no standard in the Life cycle orchestration (LSO) specification from MEF or from ETSI [93], which can transfer any SFC isolation or confidentiality attributes to the VNF from the VNF Manager.

Because the research objective is to ensure confidentiality and integrity in an SFC, this research is affected by all aforementioned SFC and VNF challenges. Correspondingly, there is a need to abstract and simplify this security objective. It is no research which has modelled these security requirements, but there have been many attempts to model the relationships between the network layers and to model the interconnection points of NFV in general. Quing et al. [126] show that there is a relationship between the different network layers where they defined the relationship as a multidimensional cube. This network model can be interpreted as a clear indication of a vertical and horizontal security dependency.

The modelling of interconnection points between NFV domains are reflected through research projects funded under the Framework Project 7 (FP7) and the Horizon 2020 program such as METIS [127], MCN [128] , CONTENT [129], T-NOVA [63], UNIFY [64], 5G-NORMA [130] and SONATA [131]. These projects chal-

lenge the less defined parameters of the ETSI model and mainly focus on network orchestration. Most vendors, standardisation organisation and related research define this multi-domain network orchestration as a part of the ETSI VIM framework [132]. Hence, the network orchestration models mainly reflect how the VIM controls the network and implicitly how the network control plane is coordinated among the different network domains.

It is assumed that the network control is profoundly affected by how the underlying network protocols are interconnected and stacked. However, from a data plane perspective, the SFC can either be provisioned in a virtual overlay network or it can be stitched together by two different network domains (Chapter 2.6). The stitching of the SFC is known to lack flexibility and to be operational expensive [133], but with automated network control and separation, it has the potential of being more efficient than multiple overlay networks. This is due to the packet overhead. These two concepts reflect the two strategies of interconnecting multiple SFC domains. This is previously defined as stratification strategies (Chapter 2.6), which is reflected by using overlay networks and horizontal multi-domain interconnections. However, the modelling of the network is a matter of perception, where different researchers and organisations have different approaches to controlling multiple network domains. From a network control plane perspective, the models are often referred to as the NFV and the SDN approach. This is exemplified by Medhat et al. [134] and López et al. [135].

This separation of the network control methods has a history from the standardisation attempts from ONF (TS-027 [136]), IETF (SFC Control Plane [137]) and ETSI (GS NFV-MAN [93]). Historically, the SDN method was aiming to control only one network domain with a focus on the southbound interface (SBI). Here, the east-west flow synchronization between controllers was mainly achieved by the use of SDNi [138] or the Border Gateway Protocol (BGP) [35]. However, SDNi is not suitable between different operational domains and BGP had limited capabilities to control advanced traffic flows. When researchers were faced the aforementioned research challenges of controlling an SFC across multiple operational domains, they addressed this challenge by supporting multiple southbound SDN protocols and multiple APIs on the northbound side of the SDN controller. Enabling the application plane in the SDN controller, with northbound interfaces, made it possible to use a hierarchy of SDN controllers and having top-level network orchestrators [134]. This resulted in more standardised northbound APIs, similar to the objectives of ETSI. It also resulted in synchronized control planes between the multiple domains allowing them to be perceived as one controller from the NFVO perspective. The API interfaces enabled an abstraction of the network control in the SDN controller. This abstraction layer translates different control

plane protocols into one standard in a vertical manner such as the Forwarding and Control Element Separation protocol (ForCES) [37], the Control Orchestration Protocol (COP) [98] and the Application-Based Network Operations (ABNO) [139]. The downside of an abstraction of network control, is that an abstraction can potentially result in a lack of information. Related to the research objectives in the thesis, the current models indicate a lack of a security abstraction, in particular, SFC confidentiality and integrity. This thesis aims to address this problem by investigating the low-level SFC forwarding methods (close to the SBI) in order to identify the attributes needed for making network abstractions.

3.2 SFC forwarding methods (RQ-2)

It was mentioned in the previous section that Munoz et al. [98] have identified different SFC technologies and that Hantouti et al. [99] point out that there are many research challenges related to SFC orchestration, such as advanced SFC routing. Most importantly, there is arguably a need for abstracting metadata about the confidentiality and integrity in an SFC. In order to extract this metadata, this research aims to identify the attributes in the existing data plane technologies. Correspondingly, this research aims to evaluate the existing SFC technologies towards these research challenges, which in particular includes investigating how secure the current forwarding mechanisms are and to survey how related research articles aims to solve SFC forwarding on the data plane. This reflects research question 1 and 2, which aim for identifying the security gaps and finding the security requirements. The related work is split into four categories; (1) SFCs by overlay tunnels, (2) the flow-based approach, (3) stateless segment routing and (4) stateful SFCs.

1 - A full meshed tunnelled overlay network is an "alternative method" of creating an SFC. It is possible to set up all the Virtual Links between every VNF with tunnels (i.e. IPsec [140], LISP [141], GRE [142] or VXLAN [143]). Nakamura et al. [144] suggested such a method by proposing a "Protocol independent FIB architecture for network overlays". However, this approach is not investigated, while it is assumed that the lack of SFC state makes it challenging for the VNFs to choose the correct SFC path. Additionally, the complexity of managing a very high number of tunnels is also assumed to be challenging. However, it is considered as a research opportunity to investigate if the tunnels can be used on an aggregated level, where the tunnels can encapsulate SFC state data [145], such as NSH.

2 - Another method of supporting a data plane SFC is what this thesis refers to as the flow-based approach. This method adopts the principles from SDN OpenFlow and BGP Flowspec [29] and aims to enable SFC forwarding based the attributes of the originating data packet. This implies identifying an SFC-flow based on layer 2, 3 and 4 only. The flow-based approach was clearly represented in the

"STEERING" method by Zang et al. [146]. However, the method lacks attributes to differentiate the SFCs from each other. Blendin et al. [147], Ding et al. [148], Abujoda et al. [149] and Zave et al. [150] aimed to reinterpret existing packet headers and to share SFC information, such as the L2 Ethernet MAC address, the IP option field, the IP Differentiated Services Code Point (DSCP) field or the TCP session field. However, their methods lack the capability of identifying a flow when for example an SF or an SFF change the packet headers (i.e an HTTP proxy). Qazi et al. [151] aimed to solve this problem by the SIMPLE method by processing and correlating the packets in front of the VNF, but the method is complex and the accuracy is questionable. Hence, the need for any additional packet header information about the SFC emerged.

In 2014, Quinn et al. [152] introduced the NSH as a data plane packet header which can contain information about the SFC data path. The protocol is transport independent, meaning that the NSH header can be encapsulated by i.e Ethernet, IP, VXLAN-GPE [39], NVGRE [153] or UDP. In the same year, H. Zhang presented the Service Chain Header [154]. This draft also introduced the roles and components for the SFC data plane forwarding. These concepts were abstracted by J.Halpern et al. [50] that in 2016 followed up by not adapting the protocol itself, but adopting the concepts of an abstract framework for SFC forwarding, which is the current SFC standard. However, since 2013, there have been many attempts for adapting to the SFC framework [155]. There have historically been two research tracks, namely stateful SFC headers and stateless SFC by segment routing.

3 - Stateless segment routing of SFCs has primarily been driven by the IETF, where they aimed for utilising MPLS [52] and IPv6 [53]. The method does not require any additional underlay network, and it has therefore been attractive for telcos in the 5G domain [156]. Due to its' lack of state capabilities and the need for a uniform communication protocol across different domains (Chapter: 2.3), it can be challenging to integrate and to achieve flexibility. However, this can be solved by a flexible control plane (Section: 3.3). By August 2019, the IETF has endorsed the transport-independent SFC encapsulation scheme of NSH, and they are also encouraging a combination of both segment routing and NSH headers [59].

4 - For stateful SFCs, there have been attempts to overcome the overlay constraint and the overhead in the NSH header. Jacquenet et al. [157] suggested to include an SFC header in the IPv6 extension header. While, in 2018, Hantouti [108] suggested to put a compact SFC header encoded in the layer 2 frame. However, most recently, the related research of stateful SFCs involves utilisation of the NSH protocol in different application domains, such as G. Li et al. [158] suggested for optical network. The most related work to the contributions in this thesis, concerns multi-domain NFV topologies and packet isolation in SFCs. Here, Kulkarni

et al. [159] pointed out scaling issues in NSH in multi-domain topologies. They suggested to scale down NSH, by the use of what they named Neo-NSH. Later, IETF released an RFC in 2018 of hierarchical SFC RFC in 2018 [160] which implicitly confirmed the studies from Kulkarni et al. Similarly, in 2019, Hantouti et al. [161] state the benefits of hierarchical SFCs by mentioning scaling, granularity and management. Hierarchical SFCs was originally designed to allow the decomposition of network architecture into multiple sub-domains. However, this research in this thesis is aimings to adopt the hierarchical principle into creating granular security domains. Further, from an application domain perspective, this research also aims for creating security function connected to NSH. This is similar to what Mehmeri et al. [121] suggested in 5G networks. This research in this thesis aims to extend this with security functions in order to support granular isolation and encryption. Note: The RFC publication of hierarchical NSH [160] was published after research article 3 in this thesis, which also introduced hierarchical NSH. However, the research objective, the content and the application domains in these publications were different to the work in this thesis.

As a reminder of the research challenge; For all these methods it is possible to use one global underlay network, such as an interconnected MPLS EVPN [162] or a Software-Defined Wide Area Network (SD-WAN) with IPsec [163], in order to interconnect the data centre resources into one network domain (Chapter: 2.2.2). However, it does not solve the SFC forwarding challenge of integrity and isolation.

Openly sharing SFC information in the data packets rises a general security concern since this metadata contains sensitive information, which can be modified by intercepting actors. Hence, neither the use of a list of segments nor SFC headers is perceived as secure. This questions the integrity of the NSH packet. The IETF NSH working group has specified an integrity check mechanism [164] to ensure that NSH headers cannot be modified. This NSH header verification was limited to NSH header integrity check only and did not include data encryption. However, the draft is currently expired and the status of the working group [164] is not known. Similarly, Wang et al. [122] have suggested Secure In-Cloud Service Function Chaining (SISC) in order to protect the SFC header by header anonymisation. However, it does not solve the data encryption or integrity problem.

With respect to packet encryption schemes, a very promising research is the Secure EVPN contribution by the IETF [162]. The publication in July 2019 suggests IPsec encryption of layer 2 frames between interconnected data centres. However, the publication does not yet discuss network isolation of the SFCs.

3.3 Multi-domain control plane architectures (RQ-3)

Research question 3 aims to suggest an architecture which is able to ensure the integrity and confidentiality of an SFC. For an SFC packet, this implies to define a new packet header structure which can be abstracted into a high-level multi-domain architecture. Based on the aforementioned SFC forwarding limitations in the NFVO, a multi-domain network control plane gives opportunities which the ETSI NFV components do not support. Hence, the related work of network control plane architectures is discussed.

Kulkarni et al. [159] show that NSH does provide an independent service plane and it enables metadata of the VNFs to be exchanged by tagging specific NSH packets with security attributes. Therefore, it is assumed that all stateful SFC packet headers, such as NSH, can be used in combination with an encrypted network underlay, such as IPSec channels [20]. However, a research challenge is how a network control plane can control security features in encryption applications together with traffic steering of the SFC packets.

The related work concerning these two topics of combining network control planes with NSH is categorised into two strategies:

(1) One idea is to use a distributed control plane protocol, which reflects the work from the Internet Engineering Task Force (IETF). They introduced BGP for exchanging route information for NSH [165]. They suggest using link-state capabilities in BGP, which includes distributing the location of the VNF by the use of the BGP distribution paradigm. However, the protocol does not yet contain any information about the location of the encryption or isolation capabilities. Nor does it discuss the integrity check of the BGP messages [166]. The design also requires a uniform communication mechanism, where all devices use BGP and understand NSH. However, it is assumed that the interoperability concern can be solved by using an overlay network.

(2) A different design approach is to utilise the traditional SDN concepts and apply centralised intelligence in an SDN application. The SDN concept is a very common standard in NFV deployment where, most commonly, one controller controls one administrative network. However, in a multi-domain environment, with multiple network domains of network control, the research challenge is how to coordinate the SFC flows among the different network domain. In this context, related architectures which use an SDN application as a top-level network orchestrator is discussed. The IRTF Software-Defined Networking Research Group (SDNRG) defined that SDN encompasses more than OpenFlow [167]. They also state that among the different SDN research challenges and opportunities, the

Northbound interface (NBI) in the SDN controller is a key enabler for scaling the network control [168]. Rotos et al. [90] divided the use of SDN NBIs into two categories. (2a) APIs concerning low-level flow information and (2b) APIs concerning high-level abstract network information. The first category represents the ability of an SDN controller to be vertically stacked. This includes abstracting low-level network information (from the SBI) in order to let it be managed by a top-level controller. This is represented by the Common Information Model aka the CoreModel from ONF [169]. The second category of the NBI typically includes communication towards the OSS. This research mainly focus on category one by evaluating related research on how this information is abstracted into an architecture. This includes architectures capable of abstracting information such as PCE [170], ESCAPE [171] and ABNO [139], which is now further discussed.

In 2017, Rotos et al. [90] explained how the Path Computation Element (PCE) [170] can be used in NVF environments. A PCE is a device or a server which computes the data packet path on behalf of the nodes in the network. When different administrative network domains use different protocols, a node can ask the PCE server for the path of a packet. It is most commonly used for traffic engineering in MPLS, but it has also gained attention for optical transport network. In this research, the use of PCE is most relevant as a top-level topology server in multi-domain environments. The PCE protocol (PCEP) is also a protocol which can introduce the concept of state in segment routing. However, PCE is most suitable for controlling tunnels, such as MPLS LSP tunnels or LISP, as it can aggregate traffic into these tunnels. However, PCEP can also steer SFCs as represented by Ku et al. [100], where they demonstrate how PCEP can act as an SDN orchestrator in a heterogeneous network. The PCEP is also closely related to the Border Gateway Protocol Link-State (BGP-LS) [172], which delivers network topology information to topology servers or route reflectors. Vaishnavi et al. [88] show how BGP-LS [172] can be used across different network domains in order to distribute SFCs in an SDN context. It is assumed that this approach does scale better than PCEP alone since BGP can aggregate route information.

The Extensible Service Chain Prototyping Environment (ESCAPE) [171] and ESCAPEv2 [173] architecture is based on the UNIFY architecture, which relies on a similar principle of having a centralised network orchestrator. The ESCAPE architecture exemplifies how the southbound protocol from the controller also can be a protocol such as NETconf [30], OpenFlow or generalised APIs. Note: This architectural principle was used and extended when evaluating research question 5.

Controlling a network based on Application-Based Network Operations (ABNO) [139] consist of having a top-level network orchestrator that controls the flows

in one or multiple network domains. This concept is suitable for interconnecting heterogeneous networks where the top-level network orchestration application dynamically can extract network information and make an abstract model of the topology in all subordinate network. ABNO is by design multi-domain environments where it enables network operator control applications to automatically provision network paths and access network state information.

In summary, the related work shows that SFC control plane architectures have the potential of supporting aggregated SFC information, which can be interpreted by a top-level controller. However, none of the standardised APIs or aforementioned architectures are considering the integrity and confidentiality of the SFC packets.

3.4 Applications for secure SFC (RQ-4)

Arguably, security is a major concern for operators aiming to deploy NFV services. It is not possible for the operators to protect all the users' data. However, the operators' service capabilities towards their end-users are essential for them in order to raise the level of trust from the end-users. This trust is not only reflected between the operator and the end-users, but it also indicates that the operators must trust each other. In use cases, when operators are using third party cloud services from other operators, it can be argued that these setups also conforms a chain of trusts between operators and end-users. Zhang et al. [15] confirm this statement where they show that the operators highly depend on the infrastructure providers for data security. The operators, providing the end-user services, are challenged to set requirements (i.e. Trusted Platform Module (TPM) [174]) towards the physical and virtual resources. Khondoker et al. [175] show that there are many security threats and countermeasures in NFV, where the attacks, which are seen on the platform layer (PaaS) relies upon the trust in the PaaS producers. The infrastructure cloud services (IaaS) are, on the other hand, more open for global security control on the network layer. This implies that in IaaS, both end-users and operators can have the opportunity to observe and confirm the security mechanisms, and not fully rely on trust only. Hence, this research aims to investigate the security mechanisms in the IaaS domains. Within the IaaS network domain, which in NFV/SDN is referred to as resource and network orchestration, the security challenges relate to the security protection of the data packets, the protection of the NFV components and to the protection of the VNFs. ETSI claims that this also includes protection from attacks and misconfigurations [176].

In 2017, Zu et al. [11] presented general security guidelines of NFV and SDN. They pointed out a wide range of security threats in different domains, such as the interfaces of the SDN controller and the NFV components. Out of these threats, the research in this thesis aims to investigate the security applications related to

protecting the end-user traffic. Specifically, how they ensure confidentiality and isolation in multi-domain environments. Murillo et al. [11] claimed that integrity, confidentiality, and availability are partially supported by proper access control, which is also one of the most important high-level security objective. Access control in SDN has been studied by the use of FortNOX [177] and SE-Floodlight [178]. However, Murillo et al. [11] state that FortNOX cannot handle service chaining because it does not allow multiple VNFs to perform a set of operations on the same resource. Further, Pattaranantakul et al. [179] presented an access control framework around the NFVO, which builds on the Moon framework (now terminated). Similar research contributions of cloud access control were also presented by Zou et al. [180] and Wen et al. [181], but neither of them discusses access control to VNFs in an SFC. This general lack of SFC access control in the related research was confirmed by Pattaranantakul et al. [182] and Paladi et al. [183], where they survey the security vulnerabilities in multi-cloud and multi-tenant NFV environments.

The aforementioned access control research in NFV consists of two types of access control. In this thesis, they are grouped into two types of security perceptions. One perception is access to the operators' API interfaces, while another perspective is how network functions themselves can provide security services for an end-user. These functions are named Network Security Functions (NSF). An NSF represents a normal VNF which runs a security application. Hyun et al. [123] exemplify such functionalities through their research contributions, such as providing firewall services for the end-users. A specific attribute to the NSFs, is that the end-user is indented to operate this function by an out of band interface. This means that the end-users do not have access to manage the function from the data plane, but they have to access a framework provided by the service provider. It is assumed that the background for this is that a VNF does not necessarily have an IP address, which is reachable for the end-users. Also, due to a non-bidirectional data plane [184], the VNF in an SFC can lack the ability to communicate with other VNFs. Hence, the service provider can allow the end-users to manage their service function through a dedicated interface - the Interface to Network Security Functions (I2NSF) [66]. Various research contribution [121],[123] have been published in order to utilise this interface and by abstracting security configurations by the use of NETCONF/YANG [185]. However, an abstract model of access control in an SFC is not found.

This research aimed to tackle research question 4 by utilising IPsec inside an NSF and using the I2NSF and NETConf for the management of this security function.

Running IPsec inside an NSF calls for a secure method of distributing the configuration of the IPsec tunnels. Due to the non-bidirectional SFC data plane, and

various other research challenges [186], [187], the IKE protocol in IPsec is not suitable for running inside an NSF. This has called for a set of research contributions in replacing IKE with an alternative protocol, named Software-Defined IKE (SD-IKE).

In 2017, Vajaranta et al. [188] demonstrated how SD-IKE could be used to load balance IPsec services in a cloud service environment. Their research lays the foundation of one of the research contribution in this thesis. Their contribution was adopted and extended to fit the NFV domain in order to also support access control and encryption in NFV which were used in research article 4.

Carrel et al. [189] published an IETF draft for using this concept of Software-Defined IPsec in EVPNs [162]. Currently, the draft does not support multi-tenant NFV domains and it does not discuss all the protocol attributes or where the encryption service is located.

In the SDN domain, Lopez et al. [190] published a result showing that they also have been following this research path. In 2019, a month after the publication of research article 4 in this thesis, they published a very similar work to this contribution [190]. However, their work is oriented towards the application domain in SDN controllers, while research article 4 is related to isolation and access control in multi-domain NFV. Their work is acknowledged and appreciated. Parts of their work can extend the contributions presented article 4. It also confirms the relevance of the work in this thesis.

Chapter 4

Summary of Contributions

This chapter outlines a summary of the research contributions which tackles the challenges described in the previous chapters. It shows how this work improves the security aspects of NFV by proposing a framework which supports access control and confidentiality in an SFC. First, a list of the published contributions is presented. Second, this chapter shows how the research questions are tackled by summarising the main contributions in all related articles. Lastly, a summary of all research contributions is presented. Further details about the major research contributions are presented in part II of this thesis.

4.1 List of publications

This thesis consists of five main contributions. These contributions reflect the research publications and mainly address the research questions in chronological order. (1) Gap analysis of secure interconnections, (2) requirements, (3) design, (4) implementation and (5) verification. The first two publications are published in conference proceedings, while the last three articles are published in research journals.

4.1.1 List of main publication

1 H. Gunleifsen, T. Kemmerich and Slobodan Petrovic,
An End-to-End Security Model of Inter-Domain Communication in Network Function Virtualisation, Proceedings of Norwegian Information Security Conference (NISK), Bergen, Norway, Nov 28-30, 2016 pp. 7-18,
ISSN: 1894-7735, https://ojs.bibsys.no/index.php/NISK/article/view/370

2 H. Gunleifsen and T. Kemmerich,
Security Requirements for Service Function Chaining Isolation and Encryption, IEEE 17th International Conference on Communication Technology

(ICCT), Chengdu, China, Oct 27-30, 2017, IEEE Explore, vol 4 pp. 1360-1365,
ISBN: 978-1-5090-3943-2, https://doi.org/10.1109/ICCT.2017.8359856

3 H. Gunleifsen, T. Kemmerich and V. Gkioulos
A Tiered Control Plane Model for Service Function Chaining Isolation, Future Internet 2018, vol 10(6), 46; Special edition: Software-Defined Networking (SDN) and Network Function Virtualization (NFV),
ISSN: 1999-5903, https://doi.org/10.3390/fi10060046

4 H. Gunleifsen, V. Gkioulos and T. Kemmerich,
Dynamic setup of IPsec VPNs in Service Function Chaining, Computer Networks, 2019, Elsevier, volume 160, pp. 77-91,
ISSN: 1389-1286 https://doi.org/10.1016/j.comnet.2019.05.015

5 H. Gunleifsen, T. Kemmerich and V. Gkioulos,
A Proof-of-Concept Demonstration of Isolated and Encrypted Service Function Chains, Future Internet 2019, volume 11(9), 183; Special edition: Network Virtualization and Edge/Fog Computing,
ISSN: 1999-5903. https://doi.org/10.3390/fi11090183

4.1.2 List of other publications

In addition to the main contributions, a list of contributions to other areas of research is also presented. These contributions are *not* a part of this thesis. One of these additional research contributions reflects on the relationship between SDN and military networks (Publication 6). These contributions were mainly conducted due to the research potential of applying the concepts of access control and confidentiality into military networks. A different perspective of NFV security is how end-users are aware of the lack of confidentiality and encryption among NFV service providers. Hence, a survey of the general security awareness among ISP end-users was published (Publication 7).

6 V. Gkioulos, H. Gunleifsen and G.K. Weldehawaryat,
A Systematic Literature Review on Military Software Defined Networks, Future Internet 2018, 10(9), 88; Special edition: Software-Defined Networking (SDN) and Network Function Virtualization (NFV),
ISSN 1999-5903 https://doi.org/10.3390/fi10090088

7 H. Gunleifsen, V. Gkioulos, G. Wangen, A. Shalaginov, M. Kianpour, M. Abomhara,

Cybersecurity Awareness and Culture in Rural Norway, Thirteenth International Symposium on Human Aspects of Information Security & Assurance (HAISA 2019), Nicosia, Cyprus, July 15-16, 2019, pp 110-121, ISBN: 978-0244-19096-5, https://dblp.org/db/conf/haisa/haisa2019.html

4.2 Summary of main contributions

This summary of contributions is categorised into five sections. These five sections reflect the one-to-one match of the research questions, the research method and the research publications. As mentioned previously, this research is performed in accordance with the Design Science Research Methodology, where each numbered step corresponds to one research question with one research publication. Further, the work in these studies is described by highlighting the associated research question and presenting an overview of the results obtained in the related articles.

4.2.1 The gap analysis of NFV interconnection models (RQ-1)

Research question 1:
What are the existing NFV interconnection methods, what are the security gaps within them and what is the most secure and promising network protocol for interconnected NFV-domains?

Relevant Article: *Publication 1 - An End-to-End Security Model of Inter-Domain Communication in Network Function Virtualisation (Chapter: 7)*

Contribution summary:
In this article, the different interconnection methods between NFV domains are analysed. The research shows how the different models result in different methods of setting up security trusts between NFV infrastructure domains. First, and foremost it is identified that there is a need for isolation in an SFC. Second, there also is a lack of encryption capabilities between the elements in an SFC. In summary:

- The research identified and classified different interconnection models, different standards and SFC technologies.
- It is identified a security gap of a lack of data encryption and isolation

This contribution from 2016 has later been confirmed by other studies [182, 183, 186, 90, 95] as mentioned in chapter 3. These other studies also confirm the research gap which was introduced in this contribution.

4.2.2 Defining requirements and identifying operational constraints (RQ-2)

Research question 2:
What are the security requirements and constraints when interconnecting virtual

network services between Internet Service Providers?

Relevant Article: *Publication 2 - Security Requirements for Service Function Chaining Isolation and Encryption (Chapter: 8)*

Contribution summary: In this article, it was presented a study of SFC isolation and encryption in interconnected NFV environments. First, this paper emphasised the research problem from article 1 and investigated the network-specific aspects of different types of SFC packet forwarding. Second, it presented the requirements for the problem solution and studied the constraints for securing and isolating the VNFs in an SFC. Among a list of 14 requirements and constraints, the four most important elements are listed below:

- There is a need for a new network packet header for encrypted SFCs.
- It is identified that the network control plane needs a corresponding mechanism for routing this new header.
- The new network header must be able to point to an encryption service.
- The encryption service needs an efficient key exchange protocol.

Further requirements and constraints are listed in the article in chapter 8.

4.2.3 The design of an architecture (RQ-3)

Research question 3:
Which are the required architectural components and functionalities for the enforcement of security control within interconnected network service infrastructures?

Relevant Article: *Publication 3 - A Tiered Control Plane Model for Service Function Chaining Isolation (Chapter: 9)*

Contribution summary: This article presents an architecture which introduces isolation and encryption in an SFC across multi-domain NFV environments. This paper introduces a hierarchy of SFC headers in order to solve the packet isolation problem in the network domain. In order to solve the confidentiality problem, it is suggested how the architecture can automate the setup of encrypted tunnels between the VNFs. However, the article mainly focused on the network isolation part and the architectural design. For the networking part, it is proposed to use a hierarchical control plane based on BGP. The use of BGP in each network layer is reasoned due to scaling. However, in the proof-of-concept implementation in this contribution, the experiment did only include a test of the layered packet hierarchy in a simplified static data plane. In summary, the architecture consists of:

- A specification of a new NSH header extension in order to support a hierarch-

ical data plane which supports packet isolation inside an SFC.
- A specification of a tiered control plane with the following attributes:
  Tier 1: A control plane for setting up additional control planes between multiple NFV domains.
  Tier 2: A control plane for NSH which aggregates to Tier 1.
  Tier 3: A control plane for setting up encryption with a key distribution centre dependent of the Tier 2 control plane.
- The architecture presents the overall model, the services of each component in the architecture, the protocols and interfaces between these services.
- The architecture also outlines the need for a new key distribution protocol and raise a new research challenge regarding this.

The implementation of the data plane was only conducted with static packet forwarding configuration. According to the top-level research method, the main implementation was planned in the next research contribution. Hence, this paper did not include any specific implementation of neither the network control plane nor the key distribution method during this work. However, this is raised as a research challenge for the next article in this thesis.

4.2.4 An implementation of key distribution for encrypted SFCs (RQ-4)

Research question 4:
How can a security control implementation ensure that the end-user privacy is protected in the network service infrastructures?

Relevant Article: *Publication 4 - Dynamic setup of IPsec VPNs in Service Function Chaining*

Contribution summary: This research publication describes a novel mechanism for the automated establishment of dynamic VPNs in NFV. Due to the lack of a bidirectional data plane in SFCs, this paper presents a new key distribution mechanism which can replace traditional IPsec-IKE. The contribution contains an architecture which presents a new way of utilising SDN in NFV. A bootstrapped mechanisms in a separate control plane automates the distribution of IPsec keys in a new way. The measurements show that the implementation performs better than traditional IKE in virtualised environments.

- This article introduced an Authentication Centre (AuC) for authenticating the endpoints of the Virtual Links in NFV.
- It presented the concept of virtual Security Associations in NFV, named Software-Defined Security Associations (SD-SA)
- A new architecture based on RESTconf was proposed, which was described in depth by services and interfaces.

- It was made a performance evaluation of the architecture.

This research contribution was mainly focused on the key-distribution mechanism and the automation of setting up encryption services in NFV.

4.2.5 A verification of the implemented design (RQ-5)

Research question 5:
Given the results of the previous research questions, where a suitable reference architecture is developed, how can this secure architecture be deployed and adopted within the current virtualised infrastructures and how does the deployment satisfy the security requirements?

Relevant Article: *Publication 5 - A Proof-of-Concept Demonstration of Isolated and Encrypted Service Function Chains*

Contribution summary: This article presents a comprehensive view of the developed architecture, focusing on the elements that constitute a new forwarding standard of encrypted SFC packets. It summarises the work from the previous contributions and presents the closing experimental results of how the presented implementation of the architecture fulfils the requirements defined in a use case scenario. This use case is based on the requirements defined in research article number two. This implementation, which is the fifth of the main contributions in this thesis, builds on the elements from the architecture in research article 3 and 4. Correspondingly, the networking design from article 3 was combined with the security and cloud service implementation from article 4. Additionally, an implementation of a dynamic forwarding and routing design was added in order to create a full implementation. In total, the implementation reflects an architecture for on-demand provisioning of encrypted and isolated SFC using P4, NFV and SDN architectural principles. Furthermore, this article unifies the previous research results by presenting how previous contributions inter-operate towards providing secure SFCs. The presented results highlight the capacity of encryption, access control and micro-segmented SFC in NFV. Throughout an extensive analysis of the implementation, it was successfully verified that the implementation fulfils most of the requirements defined in research article 2. The summary of the main contributions in this research article is as follows:

- A use case was created based on the requirements in research article 2. This use-case was used to verify how an implementation fulfils the security requirements of access control and encryption.
- A hierarchy of SFC headers was successfully implemented in a P4 switch, which had the ability to be dynamically controlled from a centralised network controller, by the use of RESTConf.

- The use case scenario was tested with three episodes, which included the operational aspects of provisioning, resilience and integrity. Further, it was verified that the implementation had fulfilled the requirements which were defined previously.

This final implementation of the architecture proposed to use RESTconf between the network controller and the virtual switch. This is a different technology to what proposed in the architectural article (Article 3). This was a practical decision due to a lack of suitable development frameworks. It is arguable that the underlying communication protocol of SFC route distribution is not of great importance of the research contributions. The most important element of research article 3 is the architecture which reflects the interaction between the components. This has not changed. However, it is of great interest in future work to evaluate how different routing protocols affect the performance of the architecture. It is also discovered that there are elements in the contribution which have room for performance improvements. Chapter 5 elaborates about these concerns.

4.3 Thesis research contribution

The main objective of this research, as stated previously (Chapter: 1.3), was to provide a mechanism which ensures the confidentiality, integrity and availability of the end-users' NFV traffic. This problem was tackled by dividing it into five research questions, which resulted in five main research contributions. In sum, the thesis contribution consists of this collection of the aforementioned contributions. In total, the final research contribution is proposing an architecture which supports confidentiality and access control by providing network isolation and encryption. This is supported by the final proposal of a security framework, which includes a centralised network controller with coordinated authentication and authorisation mechanism of the SFC elements in multi-domain NFV environments. The isolation part is achieved by using a hierarchy of SFC headers, while the confidentiality part is achieved by using distributed IPsec tunnels. By successfully following the DSRM method, the research challenges were tackled by iteratively improving the architectural design throughout a consecutive line of contributions. In the end, this resulted in an implementation following modern technology trends such as RESTconf and P4, which was verified towards a set requirements. In total, this research effort is perceived as an incremental, but important, contribution towards access control and confidentiality in NFV.

Chapter 5

Limitations and recommendations for future work

This chapter includes an evaluation of the limitations of the published articles and general recommendations for future work. Accordingly, this chapter evaluates how the collection of research contributions have fulfilled the answers to the research questions in this thesis. Further, this thesis seek to propose research directions which can enhance the existing results in the research publications in this thesis.

This chapter is organised into 5 sections which each discusses future work and research limitations for each research article respectively.

5.1 Evolution of NFV models (RQ-1)

In the first research contribution (Article 1), one of the research objectives was to survey the existing interconnection models and technologies. This was performed in order to identify if access control and confidentiality were supported in multi-domain NFV environments. This survey did not identify any solution for the problem while it published a brief taxonomy of the relevant standards and technologies. Due to a rapid growth in research contribution in the field of NFV and SFC, the number of technologies, contributing vendors and architectural models have evolved since this contribution.

This reflects a general observation, where it is noticed that the research articles in this thesis are connected to a research community which is evolving relatively fast. These research contributions have been conducted during a period of four years, which reflects how the early publications get less relevance during the years. However, even if the technologies in the research community has evolved, the research results and their conclusions are still valid. Mainly, it is the related work which gets outdated. This is reflected throughout all of the research articles in this thesis. Hence, this thesis aimed to compensate for this lack of related work in the

research articles by presenting an extensive chapter of related work in this thesis (Chapter 3).

Further, there are elements in the research articles which in retrospect is evaluated as less precise. There are elements, such as naming conventions, which have been affected by standardisation organisations, new technologies, new upcoming standards and consolidations of vendors, organisations and research projects. This is addressed in chapter 2 and 3.

The most important limitations in Article 1 is listed as follows:

- The article lacks an up-to-date taxonomy of the related work. Most importantly, the most recent technology of segment routing with NSH [59] is not considered.
- The article also lacks an up-to-date taxonomy of the current research challenges, which is addressed in chapter 3.
- It is noticed that the simplified model of horizontal and vertical security is open for misunderstanding. It does not clearly show that there can be multiple control planes for different data planes and service planes. However, the main objective of the model is still valid as it shows the importance of reflecting the high-level abstract security objective into a data plane protocol, which fulfils the abstract VNFFG definition.
- In the NFV technology table in Article 1 (Table: 7.2) there is a clear opening for misunderstanding of the encryption column for the control plane. Here, the objective of this column is simply to address whether the control channel can be encrypted or not.

For future work, this work of identifying research challenges within the security and access control in NFV is considered as a continuous task. However, in this research article, it is claimed that Block-chain has a research potential for solving the chain of trust problem in the NFV model. Recently, Rebello et al. [95], demonstrated that Block-chain could resolve the problem of both vertical and horizontal trust between the components. This research contribution solves a part of the research problem, which is not investigated in this thesis. However, their trust model was only set for the orchestration plane and was not reflected on the data plane or in SFC orchestration component.

5.2 Consolidation of requirements (RQ-2)

The second publication (Article 2) presents the requirements and constraints for the architectural components and their functionalities in order to support microsegmented SFCs. Hence, this article clearly replied to the research question. However, the requirements which were specified in the publication were very gener-

alised to all interconnection models. During the development of the architecture, additional requirements were raised, which were added in the fifth article. The list of limitations in the requirements specified in this second research contribution is as follows:

- In this article, it is listed 9 requirements and 5 constraints. However, in the verification paper (Article 5), it is only referred to 8 requirements. It is arguable that the fifth research contribution addressed all these requirements, but three of the requirements in Article 2 were not explicitly discussed in Article 5. These requirements were related to dynamic tunnels, encryption flexibility and overlay networks. The overlay network requirement is not discussed, because it is a very fundamental functional requirement for the tiered architecture. Similarly, dynamic tunnels is also a very fundamental requirement which is included in the other requirements of hop-by-by encryption, isolation and key distribution. Neither was the flexibility in encryption types discussed. However, it can be claimed that this requirement is fulfilled by the virtual architecture itself. Since the architecture was based on an IaaS infrastructure with *virtual* encryption services, the encryption service is by design flexible. Additionally, due to a consolidation of other related work [183, 90, 95], the naming conventions of the requirements are redefined in order to accommodate and consolidate to more generalised requirements. For example, instead of requiring "a new east-west communication channel", the requirement is renamed to a more generalised term, namely "control plane flow distribution".
- In this article, it is also raised the need for more specific requirements. However, due to the findings in research article 3 and 4 , the requirement list in Article 5 was altered to be more generalised in order to support different architectures and network abstractions.
- During the work with the architecture in Article 3, 4 and 5, it was also discovered additional requirements which should have been claimed in this article. Scaling and performance are major concerns. This was addressed by making performance tests in Article 5, but such requirements are questionable if they should be included in a requirement list. This is because performance demands are evolving over time and are very difficult to define.
- The figure which explains the packet encryption is not clear (Chapter: 8.2). The underlying protocols were omitted and it is not clear why some packet examples are not demonstrating encryption. This figure was updated in the related work chapter in this thesis (Chapter: 3).

5.3 Architectural iterations (RQ-3)

In the third research contribution, which proposes an architecture (Article 3), the research objective was to suggest an architecture which supported the requirements from the previous research contribution. It is claimed that the architecture supports the requirements, which is provided evidence for in the final research contribution (Article 5). However, this third contribution is not fully aligned with Article 5. The list of differences and limitations is here presented:

- In this article, it was suggested to use BGP as a distribution protocol for the BGP Link States in the VNFs. This is a method which in retrospect is considered challenging to get standardised. However, it can be claimed that the selection of the routing protocol does not affect the general architecture. In research article 4 and 5 this is changed from BGP to RESTconf. It is assumed that RESTconf is easier to adapt to different schemes of routing protocols among operators. On the other side, it is questioned if RESTconf is capable of scaling in larger environments. A RESTconf approach clearly requires a centralised control centre which can monitor configuration changes. Hence, it is of great interest to measure how different routing protocols can support a hierarchy of SFC headers.

5.4 Lack of security features (RQ-4)

The fourth research article contains an implementation of a key distribution mechanism. The related research question indicated a full implementation of the architecture, but only a subset of the overall architecture was presented. This lack of implementation is responded to by presenting a full implementation in Article 5. This article was instead fully devoted to the key distribution implementation in the overall architecture. Implementation specific details resulted in a more extensive description of the architectural design, which naturally extends the capabilities in the original architecture presented in Article 3. While evaluating this research contribution, it is discovered that the architecture has room for improvements.

- It is noted that Lopez et al. [190] published a similar research contribution a month after article 4 was published, which points out some attributes which was not consider. It was observed that there was a lack of a Diffie-Hellman mechanism in the key distribution scheme of SD-SA. This is a definite improvement in the distribution of the keys. Research article 4 has limited the research scope by assuming that the control channel by RESTconf was secure, and therefore it was also assumed that no further protection by key derivation was needed. However, it is realised that even if the RESTconf channel is secure, the keys should also be protected from other entities, such as malicious

instances of the VNFs.
- Compared to the previous research publication (Article 3), it was made a slight change in the top-level key distribution architecture. The original architectural design suggested a mechanism based on KINK. However, it was discovered that KINK did not support IKEv2 and that the KINK endpoint addresses are not easily distributed by the KINK protocol. Due to this lack of features in KINK, the architecture was modified in this research article (Article 4). This architectural change reflects the iterative approach in the DSRM research method.

5.5 Evaluation of the final result (RQ-5)

The fifth research contribution aimed to verify how the implementation of the architecture fulfilled the aforementioned requirements. This research contribution answered to this by presenting an extensive evaluation by a use case with different scenarios which reflected the requirements. Most of the requirements were fulfilled, but there is, in particular, one requirement which is only partially solved. Additionally, it is identified that there are security implications which go outside of the architecture which is not considered. In total, the list of evaluations is as follows:

- The performance of the architecture is questionable. This is also one of the requirements which was not fulfilled. However, it is excepted that hardware accelerators for P4 potentially can solve this problem.
- Through the executed studies of encrypted SFCs, it is identified that the Quality of Experience (QoE) from an end-user perspective is a crucial element for NFV adoption. The results have a great potential for improvement both regarding hardware accelerators and integrated QoS.
- Another part of access control, which is not studied, is how abstracted security policies for access control can be structured in the NFVO domain. It is assumed that the orchestration system should be able to check for invalid security policies defined by end-users or operators.
- It was only tested a few very simple packet injection mechanisms. Hence, it is highly suggested to investigate the architecture for more security vulnerabilities. For example, how replay attacks and general packet crafting can result in misconfigurations or a breach in the access control.

Chapter 6

Conclusions

Securing a multi-domain NFV environment from eavesdropping, and supporting access control by isolation, is a task intertwined with a multitude of challenges. The challenges arise both from the characteristics of the components that constitute the network, and the attributes in the network packets. During these studies, this has been examined by aiming for the attainment of providing a framework which supports this security goal. Within this context, these studies were targeted towards both a data packet forwarding mechanism and a key distribution scheme in the NFV environments. The studies were conducted in five consecutive steps, where the research contributions identified the security challenges, set the requirements, made an architecture, implemented the architecture and verified it towards the requirements.

Accordingly, the first two results of these studies highlight the security challenges and requirements for confidentiality and fine-grained access control in SFCs. The different interconnection methods were analysed and showed how trust between orchestrating NFV components defines the fundamental security relation in any interconnected NFV environment. It was identified that the high-level abstract definition of a distributed security policy in NFV is crucial for the underlying security features. Further, the lack of security policies for isolation and confidentiality is reflected in all abstraction layers. Consequently, the incorporation of the information flow between the horizontal cross-domain and the vertical cross-NFV layers is essential. The low-level description of the SFC was specifically targeted, aiming for defining the security requirements for isolation and confidentiality within these chains. The different security aspects of the different abstraction layers were surveyed and most importantly, the underlying packet forwarding schemes were analysed. This research identified a set of security attributes which reflects the security objective of access control and confidentially. The extraction of these security requirements contributes to an alignment of research challenges, a direction

of further research and it accommodates a base for implementation verification.

Further, these studies have proposed an architecture which incorporates the aforementioned requirements of isolation and encryption in an SFC. Within these studies, it is listed a set of service components and interfaces which supports a hierarchy of interconnected components and control planes. This is accommodated by a multi-layered SFC header which enables isolation of SFC packets. The network control of these SFC packets is reflected in a hierarchical control plane architecture. Further, it is also proposed a virtualised encryption service, which distributes encryption keys by a new protocol customised for NFV environments.

Finally, within these studies, an implementation of the architecture was tested towards the aforementioned requirements. It was created a use case with three episodes which reflected the requirements. By using overlay networks, virtualisation technologies and programmable switches, the architecture was implemented in a customised NFV environment. It was successfully verified that the architecture supported the different scenarios in the use case. This final result summarises the work from the previous contributions and also summarise the work of this thesis.

These research contributions incorporate a small part of the NFV security challenges in general. However, the contributions open up for further investigation of access control and isolation in NFV. This is not only referring to security, but also other aspects of NFV such as management and operation. Service provisioning, end-user quality of experience, packet forwarding optimisation and standardisation of the abstract security policies are all important research areas, which are essential to the deployment of a fully functional NFV environment supporting isolation and confidentiality.

References

[1] Mahdi Daghmehchi Firoozjaei et al. ‘Security challenges with network functions virtualization’. In: *Future Generation Computer Systems* 67 (2017), pp. 315–324.

[2] Gianpietro Lavado. *Why it is time to get serious about NFV, Whitestack webpage*. Available online: https://www.whitestack.com/posts/serious-about-nfv/ (accessed on 01 August 2019). 2019.

[3] Antonio Pietrabissa et al. ‘Requirements and Use cases system for Virtualized Network Functions platforms’. In: *Journal of Telecommunication System and Management (JTSM)* (2014).

[4] Marco Forzati, Claus Popp Larsen and Crister Mattsson. 'Open access networks, the Swedish experience'. In: *12th International Conference on Transparent Optical Networks (ICTON)*. 2010, pp. 1–4.

[5] Paul Voigt and Axel Von dem Bussche. 'The eu general data protection regulation (gdpr)'. In: *A Practical Guide, 1st Ed., Cham: Springer International Publishing* (2017).

[6] ETSI. *Network Functions Virtualisation (NFV) in ETSI*. Available online: https://www.etsi.org/technologies/nfv (accessed on 01 August 2019). 2019.

[7] ETSI. *Network Function Virtualization (NFV) Use Cases 001 v1.1.1*. Available online: http://www.etsi.org/deliver/etsi-gs/nfv/001_099/001/01.01.01-60/gs-nfv001v010101p.pdf (accessed on 04 June 2019). 2013.

[8] ETSI. *Network Functions Virtualisation (NFV) NFV-SEC 001 Problem Statement*. Available online: http://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/001/01.01.01_60/gs_NFV-SEC001v010101p.pdf (accessed on 04 June 2019). 2014.

[9] Ken Peffers et al. 'A design science research methodology for information systems research'. In: *Journal of management information systems* 24.3 (2007), pp. 45–77.

[10] Bill Kuechler and Vijay Vaishnavi. 'On theory development in design science research: anatomy of a research project'. In: *European Journal of Information Systems* 17.5 (2008), pp. 489–504.

[11] Shao Ying Zhu et al. *Guide to Security in SDN and NFV: Challenges, Opportunities, and Applications*. Springer, 2017.

[12] Keith Foote. *A Brief History of Cloud Computing, Datversity webpage*. Available online: https://www.dataversity.net/brief-history-cloud-computing/ (accessed on 01 August 2019). 2017.

[13] Amazon. *What is Cloud Computing, Amazon webpage*. Available online: https://aws.amazon.com/what-is-cloud-computing/ (accessed on 01 August 2019). 2019.

[14] Anthony D Josep et al. 'A view of cloud computing'. In: *Communications of the ACM* 53.4 (2010).

[15] Qi Zhang, Lu Cheng and Raouf Boutaba. 'Cloud computing: state-of-the-art and research challenges'. In: *Journal of internet services and applications* 1.1 (2010), pp. 7–18.

[16] Jeff Boles. *Introduction of HIC*. Available online: http://www.infostor.com/storage-management/hyperconvergence-next-generation-virtualization.html/ (accessed on 01 August 2019). 2012.

[17] Nutanix. *Hardware platforms, Nutanix webpage*. Available online: https://www.nutanix.com/products/hardware-platforms (accessed on 01 August 2019). 2019.

[18] *Openstack website*. Available online: https://www.openstack.org/ (accessed on 04 June 2019). 2019.

[19] Thomas Kemmerich, Vivek Agrawal and Carsten Momsen. 'Chapter 10 - Secure migration to the cloud—In and out'. In: *The Cloud Security Ecosystem*. Ed. by Ryan Ko and Kim-Kwang Raymond Choo. Boston: Syngress, 2015, pp. 205–230.

[20] Sheila Frankel and Suresh Krishnan. *IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap*. RFC 6071. Feb. 2011.

[21] Nick Feamster, Jennifer Rexford and Ellen Zegura. 'The road to SDN: an intellectual history of programmable networks'. In: *ACM SIGCOMM Computer Communication Review* 44.2 (2014), pp. 87–98.

[22] Diego Kreutz et al. 'Software-defined networking: A comprehensive survey'. In: *arXiv preprint arXiv 1406.0440* (2014).

[23] Open Networking Foundation (ONF). *OpenFlow Switch Specification version 1.5.1 (TS-025)*. Available online: https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf (accessed on 04 June 2019). 2015.

[24] Sushant Jain et al. 'B4: Experience with a globally-deployed software defined WAN'. In: *ACM SIGCOMM Computer Communication Review*. Vol. 43. 4. ACM. 2013, pp. 3–14.

[25] Lucien Avramov and Maurizio Portolani. *The Policy Driven Data Center with ACI: Architecture, Concepts, and Methodology*. Pearson Education, 2014.

[26] Kenneth Tam et al. *UTM Security with Fortinet: Mastering FortiOS*. Newnes, 2012.

[27] Tony Sangha and Bayu Wibowo. *VMware NSX Cookbook*. Packt Publishing Ltd, 2018.

[28] Michael Smith et al. *OpFlex Control Protocol*. Internet-Draft draft-smith-opflex-03. Work in Progress. Internet Engineering Task Force, Apr. 2016.

[29] Pedro R. Marques et al. *Dissemination of Flow Specification Rules*. RFC 5575. Aug. 2009.

[30] Rob Enns et al. *Network Configuration Protocol (NETConf)*. RFC 6241. June 2011.

[31] Fortinet. *SDN Integration*. Available online: https://www.fortinet.com /solutions/mobile-carrier/physical-hybrid-vnf/sdn-integration.html/ (accessed on 01 August 2019). 2019.

[32] Open Networking Foundation (ONF). *Functional Requirements for Transport API (TR-527)*. Available online: https://www.opennetworking.org/wp-content/uploads/2014/10/TR-527_TAPI_Functional_Requirements.pdf (accessed on 04 June 2019). 2016.

[33] Rob Shakir et al. *gRPC Network Management Interface (gNMI)*. Internet-Draft draft-openconfig-rtgwg-gnmi-spec-01. Work in Progress. Internet Engineering Task Force, Mar. 2018.

[34] Inc. Google. *gRPC Network Operations Interface (gNOI)*. Available online: https://github.com/openconfig/gnoi (accessed on 04 June 2019). 2018.

[35] Yakov Rekhter and Tony Li. *A Border Gateway Protocol 4 (BGP-4)*. RFC 1654. July 1994.

[36] Hongtao Yin et al. *SDNi: A Message Exchange Protocol for Software Defined Networks (SDNS) across Multiple Domains*. Internet-Draft draft-yin-sdn-sdni-00. Work in Progress. Internet Engineering Task Force, June 2012.

[37] Evangelos Haleplidis et al. 'ForCES applicability to SDN-enhanced NFV'. In: *Third European Workshop on Software Defined Networks*. IEEE. 2014, pp. 43–48.

[38] Bruce Davie and Jesse Gross. *A Stateless Transport Tunneling Protocol for Network Virtualization (STT)*. Internet-Draft draft-davie-stt-08. Work in Progress. Internet Engineering Task Force, Apr. 2016.

[39] Fabio Maino, Larry Kreeger and Uri Elzur. *Generic Protocol Extension for VXLAN*. Internet-Draft draft-ietf-nvo3-vxlan-gpe-06. Work in Progress. Internet Engineering Task Force, Apr. 2018.

[40] Jesse Gross, Ilango Ganga and T. Sridhar. *Geneve: Generic Network Virtualization Encapsulation*. Internet-Draft draft-ietf-nvo3-geneve-13. Work in Progress. Internet Engineering Task Force, Mar. 2019.

[41] IETF. *Network Virtualization Overlays (NVO3)*. Available online: https://datatracker.ietf.org/wg/nvo3/about/ (accessed on 01 August 2019). 2015.

[42] *VMware NSX-T Data Center Documentation*. Available online: https://docs.vmware.com/en/VMware-NSX-T-Data- Center/index.html (accessed on 04 June 2019). 2019.

[43] Aryan Taheri Monfared. 'Software-Defined Networking Architecture Framework for Multi-Tenant Enterprise Cloud Environments'. PhD thesis. Oslo University, Norway, 2015.

[44] ETSI. *Network Function Virtualization (NFV) Architectural Framework v1.1.1*. Available online: http://www.etsi.org/deliver/etsi_gs/NFV/001_ 099 /002/01.01.01_60/gs_NFV002v010101p.pdf (accessed on 23 August 2019). 2014.

[45] Chunfeng Cui et al. *Network Functions Virtualisation*. Available online: https://portal.etsi.org/NFV/NFV_White_Paper.pdf (accessed on 01 August 2019). 2012.

[46] ETSI. *Network Functions Virtualisation (NFV) 002 Architectural Framework v1.1.1.* Available online: http://www.etsi.org/deliver/etsi_gs/NFV/ 001_099/002/01.01.01_60/gs_nfv002v010101p.pdf (accessed on 04 June 2019). 2014.

[47] Dinh Khoa Nguyen et al. 'Blueprinting approach in support of cloud computing'. In: *Future Internet* 4.1 (2012), pp. 322–346.

[48] ETSI. *Network Functions Virtualisation (NFV) NFV-EVE 005 SDN Usage in NFV Architectural Framework*. Available online: http://www.etsi.org /deliver/etsi_gs/NFV-EVE/001_099/005/01.01.01_60 /gs_nfv-eve005v010101p.pdf (accessed on 04 June 2019). 2015.

[49] ETSI. *Network Functions Virtualisation (NFV) NFV-IFA 028, Report on architecture options to support multiple administrative domains*. Available online: https://www.etsi.org/deliver/etsi_gr/NFV-IFA/001_099/028 /03.01.01_60/gr_NFV-IFA028v030101p.pdf (accessed on 04 June 2019). 2018.

[50] Joel M. Halpern and Carlos Pignataro. *Service Function Chaining (SFC) Architecture*. RFC 7665. Oct. 2015.

[51] Cheng Li and Zhenbin Li. *A Framework for Constructing Service Function Chaining Systems Based on Segment Routing*. Internet-Draft draft-li-spring-sr-sfc-control-plane-framework-00. Work in Progress. Internet Engineering Task Force, June 2019.

[52] Ahmed Bashandy et al. *Segment Routing with MPLS data plane*. Internet-Draft draft-ietf-spring-segment-routing-mpls-22. Work in Progress. Internet Engineering Task Force, May 2019.

[53] Clarence Filsfils et al. *IPv6 Segment Routing Header (SRH)*. Internet-Draft draft-ietf-6man-segment-routing-header-22. Work in Progress. Internet Engineering Task Force, Aug. 2019.

[54] Ahmed AbdelSalam et al. 'Implementation of virtual network function chaining through segment routing in a linux-based nfv infrastructure'. In: *IEEE Conference on Network Softwarization (NetSoft)*. IEEE. 2017, pp. 1–5.

[55] OPNFV and OpenvSwitch community. *The Open Virtual Network*. Available online: https://wiki.opnfv.org/display/PROJ/Ovn4nfv (accessed on 01 August 2016). 2016.

[56] S Smiler et al. *OpenFlow cookbook*. Packt Publ., 2015.

[57] Anat Bremler-Barr et al. 'Deep packet inspection as a service'. In: *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*. ACM. 2014, pp. 271–282.

[58] Vitor A Cunha et al. 'An SFC-enabled approach for processing SSL/TLS encrypted traffic in Future Enterprise Networks'. In: *2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE. 2018, pp. 1013–1019.

[59] Jim Guichard et al. *Network Service Header (NSH) and Segment Routing Integration for Service Function Chaining (SFC)*. Internet-Draft draft-ietf-spring-nsh-sr-00. Work in Progress. Internet Engineering Task Force, Aug. 2019.

[60] Joel M. Halpern and Carlos Pignataro. *Service Function Chaining (SFC) Architecture*. RFC 7665. Oct. 2015.

[61] Xin Li and Chen Qian. 'The virtual network function placement problem'. In: *IEEE Conference on Computer Communications Workshops (Infocom workshops)*. IEEE. 2015, pp. 69–70.

[62] Michel S Bonfim, Kelvin L Dias and Stenio FL Fernandes. 'Integrated NFV/SDN architectures: A systematic literature review'. In: *ACM Computing Surveys (CSUR)* 51.6 (2019), p. 114.

[63] EU T-NOVA Project. *Network Functions as-a-Service over Virtualised Infrastructures*. Available online: http://www.t-nova.eu/ (accessed on 04 June 2019). 2019.

[64] EU UNIFY Project. *Unifying Cloud and Carrier Networks*. Available online: http://www.fp7-unify.eu/ (accessed on 04 June 2019). 2019.

[65] J. Garay et al. 'Service description in the NFV revolution: Trends, challenges and a way forward'. In: *IEEE Communications Magazine* 54.3 (Mar. 2016), pp. 68–74.

[66] Diego Lopez et al. *Framework for Interface to Network Security Functions*. RFC 8329. Feb. 2018.

[67] Metro Ethernet Forum (MEF). *Managed Access E-Line Service Implementation Agreement (MEF-62*. Available online: https://www.mef.net /resources/technical-specifications/download?id=112&fileid=file1/ (accessed on 04 June 2019). 2018.

[68] Artur Pilimon et al. 'Energy efficiency benefits of introducing optical switching in Data Center Networks'. In: *International Conference on Computing, Networking and Communications (ICNC)*. IEEE. 2017, pp. 891–895.

[69] 5GPPP Architecture WG. *View on 5G Architecture*. Available online: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-View-on-5G -Architecture-For-public-consultation.pdf (accessed on 04 Aug 2016). 2016.

[70] ETSI. *Network Functions Virtualisation (NFV) NFV-EVE 012 Report on Network Slicing Support with ETSI NFV Architecture Framework*. Available online: https://www.etsi.org/deliver/etsi_gr/NFV-EVE/001_099 /012/03.01.01_60/gr_NFV-EVE012v030101p.pdf (accessed on 04 June 2019). 2017.

[71] TMF. *TMFORUM APIs*. Available online: https://github.com/tmforum-apis/ (accessed on 04 June 2019). 2019.

[72] NGNM. *Next Generation Mobile Networks Alliance*. Available online: https://www.ngmn.org (accessed on 04 June 2019). 2019.

[73] 5G PPP. *5G Infrastructure Public Private Partnership*. Available online: https://5g-ppp.eu (accessed on 04 June 2019). 2019.

[74] Carlos J Bernardos et al. '5G exchange (5GEx)-multi-domain orchestration for software defined infrastructures'. In: *5GEx focus* 4.5 (2015), p. 2.

[75] 3GPP. *TR 33.899 v0.5.0, Technical Specification Group Services and System Aspects, Study on the security aspects of the next generation system*. 2017.

[76] OSM. *website*. Available online: https://osm.etsi.org/ (accessed on 04 June 2019). 2019.

[77] ONAP. *website*. Available online: https://www.onap.org/tag/nfv (accessed on 04 June 2019). 2019.

[78] Hongtao Yin et al. *SDNi: A Message Exchange Protocol for Software Defined Networks (SDNS) across Multiple Domains*. Internet-Draft draft-yin-sdn-sdni-00. Work in Progress. Internet Engineering Task Force, June 2012.

[79] OpenStack community. *The OpenStack API webpage*. Available online: http://docs.openstack.org/developer/networking-sfc/api.html (accessed on 01 August 2016). 2016.

[80] CORD. *website*. Available online: https://opencord.org/ (accessed on 04 June 2019). 2019.

[81] ONOS. *website*. Available online: https://onosproject.org/ (accessed on 04 June 2019). 2019.

[82] ETSI. *Plugtests Report v1.0.0*. Available online: https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/3rd_ETSI_NFV_Plugtests_Report_v1.0.0.pdf (accessed on 01 August 2019). 2019.

[83] Zbigniew Kotulski et al. 'On end-to-end approach for slice isolation in 5G networks. Fundamental challenges'. In: *Federated conference on computer science and information systems (FedCSIS)*. IEEE. 2017, pp. 783–792.

[84] Pat Bosshart et al. 'P4: Programming protocol-independent packet processors'. In: *ACM SIGCOMM Computer Communication Review* 44.3 (2014), pp. 87–95.

[85] *P4 Barefoot website*. Available online: https://www.barefootnetworks.com/ (accessed on 04 June 2019). 2019.

[86] P4.org. *P4 Gains Broad Adoption*. Available online: https://p4.org /p4/p4-joins-onf-and-lf.html/ (accessed on 01 August 2019). 2019.

[87] Raphael Vicente Rosa, Mateus Augusto Silva Santos and Christian Esteve Rothenberg. 'Md2-nfv: The case for multi-domain distributed network functions virtualization'. In: *2015 International Conference and Workshops on Networked Systems (NetSys)*. IEEE. 2015, pp. 1–5.

[88] Ishan Vaishnavi et al. 'Realizing services and slices across multiple operator domains'. In: *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE. 2018, pp. 1–7.

[89] Bram Naudts et al. 'Deploying SDN and NFV at the speed of innovation: Toward a new bond between standards development organizations, industry fora, and open-source software projects'. In: *IEEE Communications Magazine* 54.3 (2016), pp. 46–53.

[90] Charalampos Rotsos et al. 'Network service orchestration standardization: A technology survey'. In: *Computer Standards & Interfaces* 54 (2017), pp. 203–215.

[91] ETSI. *Network Functions Virtualisation (NFV) Management and Orchestration 001 v1.1.1*. Available online: http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf (accessed on 24 January 2019). 2013.

[92] Pedro Neves et al. ‘Future mode of operations for 5G–The SELFNET approach enabled by SDN/NFV’. In: *Computer Standards & Interfaces* 54 (2017), pp. 229–246.

[93] ETSI. *Network Functions Virtualisation (NFV) NFV-MAN 001 Management and Orchestration*. Available online: http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf (accessed on 04 June 2019). 2014.

[94] ETSI. *Network Functions Virtualisation (NFV) NFV-SEC 003 Security and Trust Guidance*. Available online: http://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/003/01.01.01_60/gs_NFV-SEC003v010101p.pdf (accessed on 04 June 2019). 2014.

[95] Gabriel Antonio F Rebello et al. ‘BSec-NFVO: A blockchain-based security for network function virtualization orchestration’. In: *International Conference on Communications (ICC)*. IEEE. 2019, pp. 1–6.

[96] George Patrick Xavier and Burak Kantarci. ‘A survey on the communication and network enablers for cloud-based services: state of the art, challenges, and opportunities’. In: *Annals of Telecommunications* 73.3-4 (2018), pp. 169–192.

[97] Ranganai Chaparadza et al. ‘Industry Harmonization for Unified Standards on Autonomic Management & Control (AMC) of Networks and Services, SDN and NFV’. In: *Globecom Workshops (GC Wkshps)*. IEEE. 2014, pp. 155–160.

[98] Raul Munoz et al. ‘Integrated SDN/NFV management and orchestration architecture for dynamic deployment of virtual SDN control instances for virtual tenant networks’. In: *Journal of Optical Communications and Networking* 7.11 (2015), B62–B70.

[99] Hajar Hantouti et al. ‘Traffic steering for service function chaining’. In: *IEEE Communications Surveys & Tutorials* 21.1 (2018), pp. 487–507.

[100] Hye-Jin Ku, JH Jung and Gu-In Kwon. ‘A study on reinforcement learning based SFC path selection in SDN/NFV’. In: *International Journal of Applied Engineering Research* 12.12 (2017), pp. 3439–3443.

[101] Gamal Sallam et al. ‘Shortest path and maximum flow problems under service function chaining constraints’. In: *Conference on Computer Communications (Infocom)*. IEEE. 2018, pp. 2132–2140.

[102] Mohammad M Tajiki et al. ‘Energy-efficient path allocation heuristic for service function chaining’. In: *21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. IEEE. 2018, pp. 1–8.

[103] Nicolas Huin, Brigitte Jaumard and Frédéric Giroire. 'Optimization of network service chain provisioning'. In: *International Conference on Communications (ICC)*. IEEE. 2017, pp. 1–7.

[104] Sahel Sahhaf et al. 'Network service chaining with optimized network function embedding supporting service decompositions'. In: *Computer Networks* 93 (2015), pp. 492–505.

[105] Peng Wang et al. 'Dynamic function composition for network service chain: Model and optimization'. In: *Computer Networks* 92 (2015), pp. 408–418.

[106] Ahmed M Medhat et al. 'Near optimal service function path instantiation in a multi-datacenter environment'. In: *11th International Conference on Network and Service Management (CNSM)*. IEEE. 2015, pp. 336–341.

[107] Andrey Gushchin, Anwar Walid and Ao Tang. 'Enabling service function chaining through routing optimization in software defined networks'. In: *53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE. 2015, pp. 573–581.

[108] Hajar Hantouti and Nabil Benamar. 'A Novel SDN-based Architecture and Traffic Steering Method for Service Function Chaining'. In: *International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT)*. IEEE. 2018, pp. 1–8.

[109] Oussama Soualah et al. 'An efficient algorithm for virtual network function placement and chaining'. In: *14th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE. 2017, pp. 647–652.

[110] Zaid Allybokus et al. 'Virtual function placement for service chaining with partial orders and anti-affinity rules'. In: *Networks, Wiley Online Library* 71.2 (2018), pp. 97–106.

[111] Deval Bhamare et al. 'Optimal virtual network function placement in multi-cloud service function chaining architecture'. In: *Computer Communications, Elsevier* 102 (2017), pp. 1–16.

[112] Marcelo Caggiani Luizelli et al. 'A fix-and-optimize approach for efficient and large scale virtual network function placement and chaining'. In: *Computer Communications, Elsevier* 102 (2017), pp. 67–77.

[113] Yu Sang et al. 'Provably efficient algorithms for joint placement and allocation of virtual network functions'. In: *Conference on Computer Communications (Infocom)*. IEEE. 2017, pp. 1–9.

[114] Chuan Pham et al. 'Traffic-aware and energy-efficient vnf placement for service chaining: Joint sampling and matching approach'. In: *IEEE Transactions on Services Computing* (2017).

[115] Sevil Mehraghdam, Matthias Keller and Holger Karl. 'Specifying and placing chains of virtual network functions'. In: *3rd International Conference on Cloud Networking (CloudNet)*. IEEE. 2014, pp. 7–13.

[116] Marouen Mechtri, Chaima Ghribi and Djamal Zeghlache. 'A scalable algorithm for the placement of service function chains'. In: *IEEE Transactions on Network and Service Management* 13.3 (2016), pp. 533–546.

[117] Milad Ghaznavi et al. 'Service function chaining simplified'. In: *arXiv preprint arXiv:1601.00751* (2016).

[118] Lucas Bondan et al. 'A framework for SFC integrity in NFV environments'. In: *IFIP International Conference on Autonomous Infrastructure, Management and Security*. Springer, Cham. 2017, pp. 179–184.

[119] Woosik Lee and Namgi Kim. 'Security policy scheme for an efficient security architecture in software-defined networking'. In: *Information* 8.2 (2017), p. 65.

[120] Konstantinos Fysarakis et al. 'A reactive security framework for operational wind parks using service function chaining'. In: *Symposium on Computers and Communications (ISCC)*. IEEE. 2017, pp. 663–668.

[121] Guanwen Li et al. 'Fuzzy theory based security service chaining for sustainable mobile-edge computing'. In: *Mobile Information Systems* 2017 (2017).

[122] Huazhe Wang et al. 'Sics: Secure in-cloud service function chaining'. In: *arXiv preprint arXiv:1606.07079* (2016).

[123] Sangwon Hyun et al. 'Interface to Network Security Functions for Cloud-Based Security Services'. In: *IEEE Communications Magazine* 56.1 (2018), pp. 171–178.

[124] Deval Bhamare et al. 'A survey on service function chaining'. In: *Journal of Network and Computer Applications* 75 (2016), pp. 138–155.

[125] Abheri Dutta. 'Achieving interoperability and novel management of Virtual Network Functions in NFV-enabled Cloud Data-centre'. MA thesis. School of Computing National College of Ireland, 2018.

[126] Qiang Duan, Nirwan Ansari and Mehmet Toy. 'Software-defined network virtualization: An architectural framework for integrating SDN and NFV for service provisioning in future networks'. In: *IEEE Network* 30.5 (2016), pp. 10–16.

[127] Afif Osseiran et al. 'Scenarios for 5G mobile and wireless communications: the vision of the METIS project'. In: *IEEE communications magazine* 52.5 (2014), pp. 26–35.

[128] MCN Project. *Funded under FP7-ICT (318109)*. Available online: http://www.mobile-cloud-networking.eu/site/ (accessed on 04 June 2019). 2019.

[129] CONTENT Project. *Convergence of Wireless Optical Network and iT rEsources iN SupporT of Cloud Services. FP7-ICT. Project*. Available online: http://cordis.europa.eu/fp7/ict/future-networks/documents/call8-projects/content-factsheet.pdf (accessed on 04 June 2019). 2019.

[130] 5G-NORMA. *5G NOvel Radio Multiservice Adaptive Network Architecture*. Available online: : https://5gnorma.5g-ppp.eu/ (accessed on 04 June 2019). 2019.

[131] SONATA. *Service Programing and Orchestration for Virtualized Software Networks*. Available online: http://www.sonata-nfv.eu// (accessed on 04 June 2019). 2019.

[132] Steven Van Rossem et al. 'NFV service dynamicity with a DevOps approach: Insights from a use-case realization'. In: *Symposium on Integrated Network and Service Management (IM)*. IEEE. 2017, pp. 674–679.

[133] Paul Quinn and Thomas Nadeau. *Problem Statement for Service Function Chaining*. RFC 7498. Apr. 2015.

[134] Ahmed M Medhat et al. 'Service function chaining in next generation networks: State of the art and research challenges'. In: *IEEE Communications Magazine* 55.2 (2016), pp. 216–223.

[135] Lorena Barona López et al. 'Key technologies in the context of future networks: operational and management requirements'. In: *Future Internet* 9.1 (2017), p. 1.

[136] Open Networking Foundation (ONF). *L4-L7 Service Function Chaining Solution Architecture (TS-027)*. Available online: https://www.opennetworking.org/wp-content/uploads/2014/10/L4-L7_Service_Function_Chaining_Solution_Architecture.pdf (accessed on 04 June 2019). 2014.

[137] Mohamed Boucadair. *Service Function Chaining (SFC) Control Plane Components & Requirements*. Internet-Draft draft-ietf-sfc-control-plane-08. Work in Progress. Internet Engineering Task Force, Oct. 2016.

[138] Hongtao Yin et al. *SDNi: A Message Exchange Protocol for Software Defined Networks (SDNS) across Multiple Domains*. Internet-Draft draft-yin-sdn-sdni-00. Work in Progress. Internet Engineering Task Force, June 2012. 14 pp.

[139] Daniel King and Adrian Farrel. *A PCE-Based Architecture for Application-Based Network Operations*. RFC 7491. Mar. 2015.

[140] Karen Seo and Stephen Kent. *Security Architecture for the Internet Protocol*. RFC 4301. Dec. 2005.

[141] Dino Farinacci and Brian Weis. *Locator/ID Separation Protocol (LISP) Data-Plane Confidentiality*. RFC 8061. Feb. 2017.

[142] Tony Li et al. *Generic Routing Encapsulation (GRE)*. RFC 2784. Mar. 2000.

[143] Mallik Mahalingam et al. *Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks*. RFC 7348. Aug. 2014.

[144] Ryo Nakamura et al. 'Protocol-Independent FIB Architecture for Network Overlays'. In: *Proceedings of the 7th ACM SIGOPS Asia-Pacific Workshop on Systems*. ACM. 2016, p. 8.

[145] Pau Capdevila Pujol. 'Deployment of NFV and SFC scenarios'. B.S. thesis. Universitat Politecnica de Catalunya, 2017.

[146] Ying Zhang et al. 'Steering: A software-defined networking for inline service chaining'. In: *21st IEEE international conference on network protocols (ICNP)*. IEEE. 2013, pp. 1–10.

[147] Jeremias Blendin et al. 'Position paper: Software-defined network service chaining'. In: *2014 Third European Workshop on Software Defined Networks*. Citeseer. 2014, pp. 109–114.

[148] Wanfu Ding et al. 'OpenSCaaS: an open service chain as a service platform toward the integration of SDN and NFV'. In: *IEEE Network* 29.3 (2015), pp. 30–35.

[149] Ahmed Abujoda, Hadi Razzaghi Kouchaksaraei and Panagiotis Papadimitriou. 'SDN-based source routing for scalable service chaining in datacenters'. In: *International Conference on Wired/Wireless Internet Communication*. Springer. 2016, pp. 66–77.

[150] Pamela Zave et al. 'Dynamic service chaining with dysco'. In: *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM. 2017, pp. 57–70.

[151] Zafar Ayyub Qazi et al. 'SIMPLE-fying middlebox policy enforcement using SDN'. In: *ACM SIGCOMM computer communication review*. Vol. 43. 4. ACM. 2013, pp. 27–38.

[152] Paul Quinn and Jim Guichard. 'Service function chaining: Creating a service plane via network service headers'. In: *Computer, IEEE* 47.11 (2014), pp. 38–44.

[153] Pankaj Garg and Yu-Shun Wang. *NVGRE: Network Virtualization Using Generic Routing Encapsulation*. RFC 7637. Sept. 2015.

[154] Cathy(Hong) Zhang et al. *Service Chain Header*. Internet-Draft draft-zhang-sfc-sch-03. Work in Progress. Internet Engineering Task Force, Dec. 2014.

[155] Mohamed Boucadair et al. *Service Function Chaining: Design Considerations, Analysis & Recommendations*. Internet-Draft draft-boucadair-sfc-design-analysis-03. Work in Progress. Internet Engineering Task Force, Oct. 2014.

[156] David Lebrun. 'Leveraging ipv6 segment routing for service function chaining'. In: *CoNEXT 2015 student workshop*. 2015.

[157] Christian Jacquenet and Mohamed Boucadair. *An IPv6 Extension Header for Service Function Chaining (SFC)*. Internet-Draft draft-jacquenet-sfc-ipv6-eh-01. Work in Progress. Internet Engineering Task Force, Jan. 2016.

[158] Victor Mehmeri et al. 'Optical network as a service for service function chaining across datacenters'. In: *2017 Optical Fiber Communications Conference and Exhibition (OFC)*. IEEE. 2017, pp. 1–3.

[159] Sameer Kulkarni et al. 'Neo-NSH: Towards scalable and efficient dynamic service function chaining of elastic network functions'. In: *20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*. IEEE. 2017, pp. 308–312.

[160] David Dolson et al. *Hierarchical Service Function Chaining (hSFC)*. RFC 8459. Sept. 2018.

[161] Hajar Hantouti, Nabil Benamar and Tarik Taleb. 'VLAN-based Traffic Steering for Hierarchical Service Function Chaining'. In: *WCNC, Marrakech, Morocco*. IEEE. 2017.

[162] Ali Sajassi et al. *Secure EVPN*. Internet-Draft draft-sajassi-bess-secure-evpn-02. Work in Progress. Internet Engineering Task Force, June 2019.

[163] Sergey Gordeychik and Denis Kolegov. 'SD-WAN Threat Landscape'. In: *arXiv preprint arXiv:1811.04583* (2018).

[164] K Tirumaleswar Reddy et al. *Authenticated and encrypted NSH service chains*. Internet-Draft. Work in Progress. Apr. 2015.

[165] Adrian Farrel et al. *BGP Control Plane for NSH SFC*. Internet-Draft draft-mackie-bess-nsh-bgp-control-plane-04. Internet Engineering Task Force, Feb. 2017. 52 pp.

[166] Paul Quinn, Uri Elzur and Carlos Pignataro. *Network Service Header (NSH)-[Review]*. RFC 8300. Jan. 2018.

[167] Evangelos Haleplidis et al. *Software-Defined Networking (SDN): Layers and Architecture Terminology*. RFC 7426. Jan. 2015.

[168] Carlos J. Bernardos et al. *Network Virtualization Research Challenges*. RFC 8568. Apr. 2019.

[169] Open Network Foundation. *Core Information Model (CoreModel)*. Available online: https://www.opennetworking.org/images/stories /downloads/sdn-resources/technical-reports /ONF-CIM_Core_Model_base_document_1.1.pdf (accessed on 01 August 2019). 2015.

[170] JP Vasseur, Adrian Farrel and Gerald Ash. *A Path Computation Element (PCE)-Based Architecture*. RFC 4655. Aug. 2006.

[171] Attila Csoma et al. 'ESCAPE: Extensible service chain prototyping environment using mininet, click, netconf and pox'. In: *ACM SIGCOMM Computer Communication Review* 44.4 (2015), pp. 125–126.

[172] Les Ginsberg et al. *BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions*. RFC 8571. Mar. 2019.

[173] Balázs Sonkoly et al. 'Multi-domain service orchestration over networks and clouds: a unified approach'. In: *ACM SIGCOMM Computer Communication Review* 45.4 (2015), pp. 377–378.

[174] Steven L Kinney. *Trusted platform module basics: using TPM in embedded systems*. Elsevier, 2006.

[175] Rahamatullah Khondoker. *SDN and NFV Security: Security Analysis of Software-Defined Networking and Network Function Virtualization*. Vol. 30. Springer, 2018.

[176] M. Chiosi et al. *Network Functions Virtualisation Introductory White Paper*. Available online: https://portal.etsi.org/nfv/nfv_white_paper.pdf (accessed on 04 June 2019). 2012.

[177] Philip Porras et al. 'A security enforcement kernel for OpenFlow networks'. In: *Proceedings of the first workshop on Hot topics in software defined networks*. ACM. 2012, pp. 121–126.

[178] Phillip A Porras et al. 'Securing the Software Defined Network Control Layer.' In: *Network and Distributed System Security Symposium (NDSS)*. 2015.

[179] Montida Pattaranantakul et al. 'SecMANO: Towards network functions virtualization (NFV) based security management and orchestration'. In: *Trustcom/BigDataSE/ISPA*. IEEE. 2016, pp. 598–605.

[180] Deqing Zou et al. 'A fine-grained multi-tenant permission management framework for SDN and nfv'. In: *IEEE Access* 6 (2018), pp. 25562–25572.

[181] Xitao Wen et al. 'Sdnshield: Reconciliating configurable application permissions for sdn app markets'. In: *46th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*. IEEE. 2016, pp. 121–132.

[182] Montida Pattaranantakul et al. 'NFV security survey: From use case driven threat analysis to state-of-the-art countermeasures'. In: *IEEE Communications Surveys & Tutorials* 20.4 (2018), pp. 3330–3368.

[183] Nicolae Paladi, Antonis Michalas and Hai-Van Dang. 'Towards secure cloud orchestration for multi-cloud deployments'. In: *Proceedings of the 5th Workshop on CrossCloud Infrastructures & Platforms*. ACM. 2018, p. 4.

[184] Joel M. Halpern and Carlos Pignataro. *Service Function Chaining (SFC) Architecture*. RFC 7665. Oct. 2015.

[185] Jinyong Tim Kim et al. *I2NSF Network Security Functions-Facing Interface YANG Data Model*. Internet-Draft draft-kim-i2nsf-nsf-facing-interface-data-model-04. Work in Progress. Internet Engineering Task Force, Oct. 2017.

[186] Linda Dunbar et al. *Seamless Interconnect Underlay to Cloud Overlay Problem Statement*. Internet-Draft draft-dm-net2cloud-problem-statement-07. Work in Progress. Internet Engineering Task Force, Feb. 2019.

[187] Markku Vajaranta, Joona Kannisto and Jarmo Harju. 'Ipsec and ike as functions in sdn controlled network'. In: *International Conference on Network and System Security*. Springer. 2017, pp. 521–530.

[188] Markku Vajaranta, Joona Kannisto and Jarmo Harju. 'IPsec and IKE as Functions in SDN Controlled Network'. In: *Network and System Security*. Springer International Publishing, 2017, pp. 521–530.

[189] David Carrel and Brian Weis. *IPsec Key Exchange using a Controller*. Internet-Draft draft-carrel-ipsecme-controller-ike-01. Work in Progress. Internet Engineering Task Force, Mar. 2019.

[190] Gabriel Lopez-Millan, Rafael Marin-Lopez and Fernando Pereniguez-Garcia. 'Towards a standard SDN-based IPsec management framework'. In: *Computer Standards & Interfaces* (2019), p. 103357.

Part II

Research articles

Chapter 7

An End-to-End Security Model of Inter-Domain Communication in Network Function Virtualisation

Published in Proceedings of Norsk Informasjonssikkerhetskonferanse (NISK), Bergen, Norway (2016): 7-18.

Håkon Gunleifsen, Thomas Kemmerich, Slobodan Petrovic

hakon.gunleifsen2,thomas.kemmerich,slobodan.petrovic@ntnu.no

Norwegian Information Security Laboratory
Norwegian University of Science and Technology, NTNU, Gjøvik, Norway

Abstract

This paper presents a new end-to-end security model for interconnected Virtual Network domains. Network Function Virtualization (NFV) has gained wide attention among Internet Service Providers during the last years. The standardization work from ETSI has outlined a common framework for Network Function Virtualization, open for multiple combinations of inter-domain communication. The communication methods consist of multiple NFV interconnection technologies and interfaces, that open up for a variety of NFV models and increased complexity. From an Internet Service Provider (ISP) perspective, the ultimate goal is to be able to freely interconnect NFV services with other ISPs in a secure and automated man-

ner. Hence, this paper presents an abstraction model of the current NFV end-to-end network transport mechanisms for inter-domain communication, to model the end-to-end security. The general work within the NFV domain is driven by multiple research contributors where academia, standardization organizations and the open-source community further develop the technology. To verify the model and contribute avoiding research silos, it is also important to classify the related research. We use the presented model for such classification of NFV interconnection mechanisms. By categorizing the differences between the NFV interconnection layers, we show that the model can be used to identify the security gap for secure network channels in NFV.

7.1 Introduction

Network Function Virtualization is based on the concept of moving Network Functions (NF) from distributed hardware (aka middleboxes), into centralized servers. Examples of such functions are DHCP server, firewalls and Intrusion Detection Systems etc [1]. When these functions are virtualized, moved off-site and centralized, the network traffic needs to be redirected to the centralized servers and also redirected between multiple instances of such functions. This paper investigates the end-to-end security of redirecting such traffic.

Security is still a major obstacle to NFV and cloud computing [2]. Network operators in Internet Service Providers (ISPs) and enterprise networks have to ensure data privacy and integrity to a high level. Lack of security features slows down the adoption rate of this technology. NFV has its origin in cloud computing where resource sharing and multiplexing are essential for the business. An important security aspect of NFV networking is isolating resources and the networks between customers. Lack of such isolation brings security risks to users and operators. The ongoing NFV standardization work shows multiple examples of how to logically separate the traffic inside an NFV Infrastructure [3]. But, when the traffic exits the administrative domain, multiple transport technologies (see Table 7.2) and types of inter-domain interfaces exist (Fig. 7.1). This makes standardization and security work challenging.

This paper addresses two challenges related to NFV interconnection and end-to-end security:

(1) Determining the types of NFV interconnection protocols that can be used in a secure environment, as well as the way how they can be classified and how they support network isolation and encryption.

(2) Explaining the dependency between the related protocols needed to set up the Virtual Network Functions, and determining whether potential protocol dependency affects the end-to-end security.

In this paper, we respond to these challenges by defining an abstract network model of protocol relations. Then we use this model to group the different interconnection protocols and show the dependency between the groups. The structure of the paper is the following. In Section 2 we give a short background of the related work. In Section 3 we introduce the necessary technical details of the NFV framework and in Section 4, we define the modelling and classification criteria of NFV interconnection methods. Then we present a four-layered security model of interconnecting NFV networks in Section 5. The technology classification is given in Section 6. Section 7 concludes this paper.

7.2 Related work

Within the research area of interconnecting NFV domains, European Telecommunications Standards Institute (ETSI) aims to lead and coordinate the work. However, private organizations, as well as researchers from industry and academia, work in parallel or in cooperation with ETSI. The research areas are split into research silos, which represents the telecommunication providers, datacenters and non-telco based cloud computing networks. The organizations that contribute to NFV inter-domain communications are summarized in Table 7.1. Their specific work on interconnection technologies is covered Section 6.

7.3 The NFV Framework

In 2012 a group of seven telecommunication operators called for research action of Network Function Virtualization (NFV) and selected the ETSI to be the home of the specification. By 2016, 290 companies, including 38 of the world's leading telco operators have joined the research group. ETSI aims to produce requirements and specification guidelines. They have published 40 documents so far, of which 7 are related to security [16] and 5 are related to network interconnection architecture [17].

One of the objectives of ETSI NFV is to separate the virtualized network functions from the physical infrastructure and management. Hence, the ETSI NFV architecture consists of three key components [17] :

- **Virtal Network Functions (VNF)** - The instances of virtualized machines

Organization	Type	Track	Contribution	Description
ETSI [1]	Std.org	Telco.	Architecture & Requirements	Top level framework
5GPP [4]	Std.org	Telco.	Architecture & Application	Study mobile standardization
IETF SFC [5]	Std.org	Telco.	Protocols & Models	Service chaining and Protocols
IRTF NFVRG [6]	Std.org	Telco.	Architecture & Requirements	NFV in general
ONF [7]	Std.org	Datacenter	Protocols & Models	OpenFlow standardization
Supercloud project [8]	Academia	Cloud	Architecture & Requirements	Self service and end-to-end security
CleanSky ITN [9]	Academia	Cloud	Architecture & Requirements	Datacenter consolidation
Mobile Cloud Netw. [10]	Academia	All	Architecture & Requirements	Datacenter for mobile networks
UNIFY [11]	Academia	Telco.	Architecture & Requirements	Unified NFV Platform
T-NOVA [12]	Academia	All	Architecture & Requirements	Open Network & business with NFV
OpenStack [13]	Open Source	Cloud	Solutions	OpenStackAPI for distributed NFV
OpenDaylight [14]	Open Source	Datacenter	Solutions	SDN platform with NFV solutions
OPNNFV [15]	Open Source	Telco.	Solutions	OpenStack platform with NFV

Table 7.1: Related research

- **NFV Infrastructure (NFVI/NVI)** - The physical and virtual infrastructure
- **NFV management and orchestration. (NFV MANO)** - Management and control of NFV

NFV brings together different industries, where each of them has their separate specifications and techniques. The NFV architecture aims to create functional blocks with interfaces to create a reference model for NVI. Hence, the ETSI models are closer to a reference model than to a protocol specification. Yet they lack information about secure channels and dependency between the components. Recently, ETSI has published documents related to reliability in order to show dependencies [18], but the dependencies with respect to end-to-end security have they not considered.

7.4 Modelling and classification criteria

We here define the deficiencies in the current ETSI model related to interconnecting NFV domains. These deficiencies are the basis for the attributes and classifiers in our new model. This section gives a background for these classification criteria.

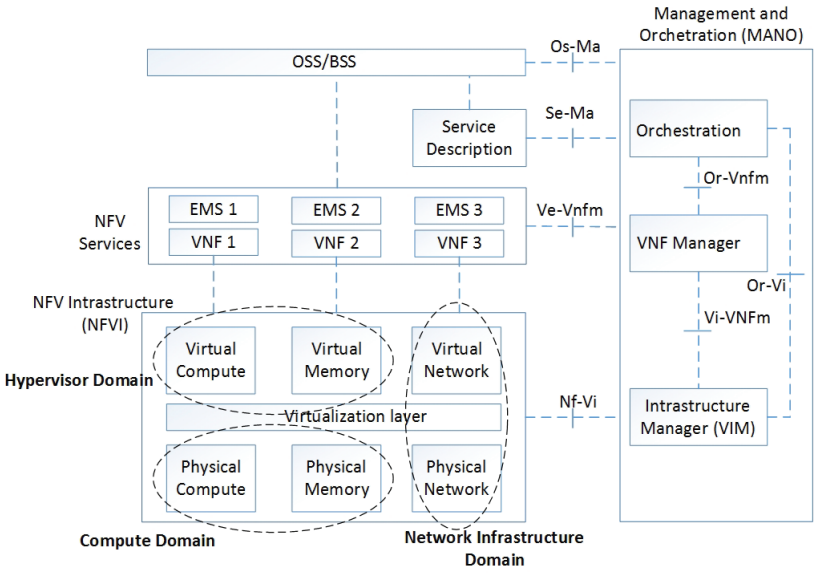


Figure 7.1: The ETSI model simplified [17]

Location of Network Control: Regarding network interconnections, the ETSI model does not specify where to run the network control. ETSI has proposed to run Software Defined Network control either in a VNF, in the Network Infrastructure domain or in the Virtual Infrastructure Manager (VIM) [19]. This implies that the ETSI model is open for multiple combinations of inter-domain communication. It is considered that network control intercommunication dependencies are not modelled by ETSI.

Network Service Chaining dependencies: According to ETSI, a Network Service is a collection of Network Functions that defines the end-to-end behaviour and forwarding paths of the Network Functions [20]. An important goal of ETSI NFV is that these forwarding paths and Network Functions are dynamically created and traffic is routed across multiple networks automatically. This means that a customer can have chained network services from multiple service providers, that are transparent to the end user. The Management and Orchestration document [20] describes the attributes of chained network services with Virtual Links and how they can be abstracted. Abstractions of end-to-end security dependencies between multiple Virtual Links are not contained in the ETSI documents.

Security dependencies between protocols: ETSI has stated many security challenges related to NFV [2], but only one document[16] discusses the security challenges of interconnecting NFV domains. [16] is a trust and topology guidance.

It does not investigate relations between NFV interconnections with respect to security levels. We claim that the end-to-end security requirements in different topologies will be different depending on the type of communication channels between NFV components (VNFC). The reason for this, is that each NFV interface should be secured and relations between them should be modelled. This is close to the protocol relation between IP forwarding and IP routing. End-user data-plane traffic (VNF-VNF) communicates on a different channel than control-plane traffic (VNFC-VNFC). The data-plane traffic flow is dependent on the management layer to configure and establish the data path. Hence, a separation of communication layers and a model of their dependencies are needed. For example: a security breach in network control or key distribution systems also invalidates the end-to-end security in end user traffic.

Traffic isolation: A key problem in virtualizing a network is that a channel is not secure if it is not end-to-end and implicitly shared.

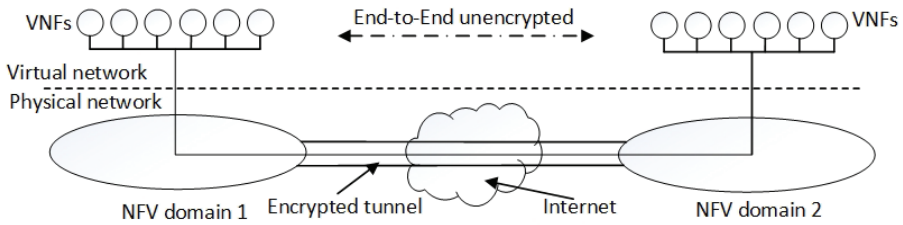


Figure 7.2: Virtualized network

When virtualizing networks, multiple overlay networks are created (Fig. 7.2). A typical solution is to create one secure channel between every NFV domain. This channel is referred to as the Virtual Link, and can potentially include management traffic as well as end-user traffic between the domains. Hence, the channel integrity is only ensured from outside of the NFV domains. One expected future scenario is that NFV will be widely adopted where most Internet Service Providers run interconnected NFV; if everyone has joined the NFV VPN, then the NFV network is considered as open and shared. Additional secure channels will therefore be needed. These channels must be secured and isolated on an individual level.

7.5 A security model of interconnecting NFV networks

To categorize the different methods to interconnect domains, this paper presents a four-layered model. The model reflects four types of communication channels

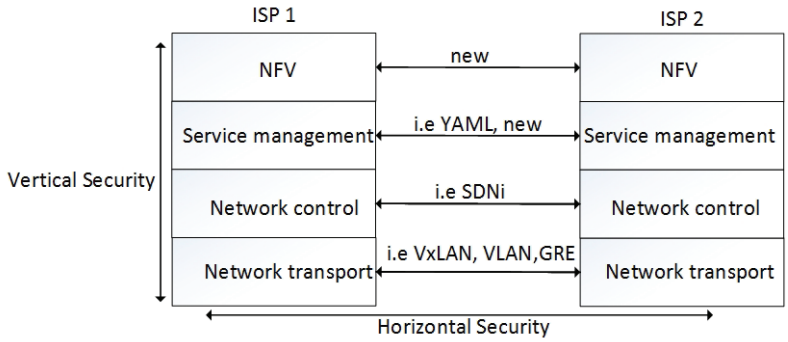


Figure 7.3: The Network Abstraction Stack

between NFV domains. It shows that the communication channels depend on each other and that they represent different levels of network abstraction. The network transport (1) is the low-level end-users communication flow between VNFs. The network control layer (2) handles topology and routing information. These need to be exchanged for the transport layer to work. The Service Management layer (3) is an abstraction of the communication needed to instantiate VNFs and Service Chains (SC) [20]; it is dependent on network control. The NFV domain layer (4) manifests a new contractual top-level end-to-end interconnection interface between NFV operators. The lower layer communication channels have one upper layer dependency, while one upper layer communication channel is responsible for multiple lower layer communication channels. For example, a network control channel can be used to configure multiple datapaths, while a specific datapath must be controlled by one master controller. The presented model reflects the fact that a security breach in the communication on the upper layers also invalidates the secure channels of all the underlying layers. Hence, end-to-end security has a new vertical aspect in addition to traditional horizontal end-to-end security (Fig. 7.3). Therefore, this dependency *must* [21] be validated for every layer. (see Section 7.6)

It is possible to add additional underlying layers to represent the physical infrastructure, or to add additional middle-layers to represent multiple levels of service abstractions. The presented model does not make any restrictions in the number of layers. It uses four layers to simplify the protocol classification and to symbolize the hierarchy and the chain of dependencies in the horizontal and the vertical axes.

7.5.1 Security Association topologies

ETSI suggests a trust guidance [16] between the NFV components, but they do not have a model of trust dependencies between the NFV components. Trust is

highly dynamic and a security framework with dynamic trust relationships must be defined. This paper suggests using the presented model to show the relationships in trust dynamics. Trust is the confidence and the reliance in the integrity of a remote entity and it is often a human decision made on an abstract level [22]. This abstract confidence in trust makes the network abstraction model suitable for modelling trust.

The model (Fig. 7.3) shows multiple layers of communication that are dependent on each other. All the layers represent different security levels and implicit trust levels. This leads to a vertical chain of trusts between the layers. Correspondingly, the communication between the ISP domains shows the horizontal chain of trust (i.e when a VNF traverses multiple ISP domains). In a chain of trust, a root of trust must be defined. ETSI has suggested that the originating VNF should have the root of trust [16]. This suggestion does not fit the model, and we claim that the root of trust must be redefined. This is because the trust of the VNF is defined before the instantiation of the VNF on the very top-level of the network abstraction stack. Then, after the VNF instantiation, the originating VNF can create consecutive trust relations with other VNFs.

This paper defines horizontal and vertical trust relations as Security Associations (SAs) similar to IKE [23] and X.509 [24] relations. The SAs are used to establish the secure channels. To ensure end-to-end security between multiple channels, our model shows that a hierarchy of SAs is needed. The root of trust is defined as the top-level SA.

Different NFV topologies make multiple combinations of the vertical and the horizontal SA axes and create multiple abstract network topologies. A common topology is a federated NFV network. A federated model implies that there is a master component in the model, which controls multiple subordinate components. This does not change the SAs in the security model, but it shows that one upper layer can have multiple SAs in subordinate layers. In federated hierarchies, sub-domains do not need to implement the whole abstraction stack. However, in an ISP inter-domain communication model all the layers will be present. In a federated sub-domain model with smaller number of abstraction layers, the highest layer in each sub-domain possesses the domain specific top-level SA (Fig. 7.4).

An intermediate model differs from a federated model. In an intermediate model, an interconnected system answers on the behalf of another interconnected system. Since the trust and security requirements differ between ISP domains, it is not possible to make horizontal trust chains without defining the end points. An intermediate model requires an additional horizontal end-to-end channel of trust. Without end-to-end SAs, it is assumed that a channel is not sufficiently secured. However

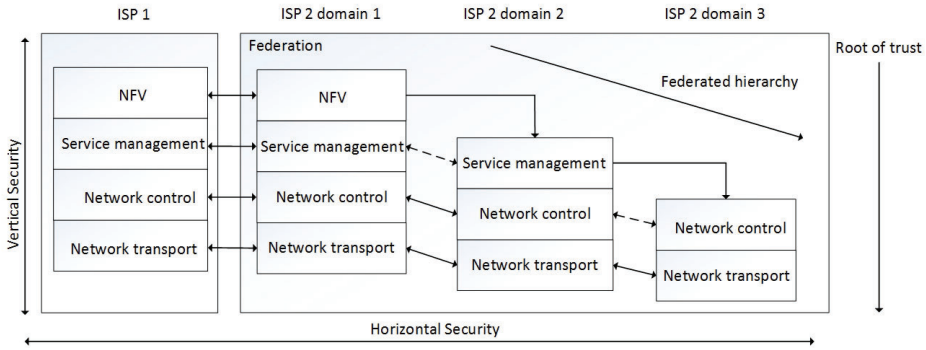


Figure 7.4: Interconnections to federated models

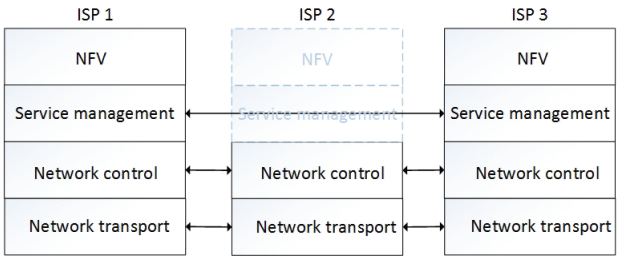


Figure 7.5: Intermediate model with trust

the network abstraction model opens up for trusted end-to-end connections as long as the highest level of abstraction has a horizontal end-to-end SA between them. The underlying layers can therefore trust the top-level SA. For the intermediate operator, a top-level one-to-one SA is needed from the origin ISP (Fig. 7.5). The distribution of the SAs is considered to be connected to the Service Graphs [25]. The Service Graph sets up the chain and the forwarding paths between the VNFs at the origin ISP. This ISP can also set up the corresponding SAs in a similar manner. This also opens up for the Service Graphs to use SA attributes in the forwarding path calculations. This allows the ISPs to define requirements and policies about secure paths, which ensures VNF integrity and confidentiality when the traffic traverses intermediate networks.

The intermediate models show the importance of secure channels, security chains and horizontal trust. Autonomous Systems of ISPs have multiple interconnection paths where a traffic path can alter between many different transit ISPs (Fig. 7.6). It is expected that some ISPs will have support for network control interaction and others will not. Therefore, the configuration of a path depends on the network transport (i.e tunnels or flows) and the network control (i.e. SDN or MPLS

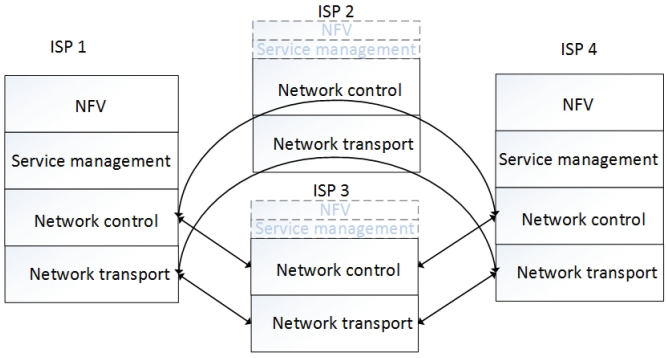


Figure 7.6: Multiple paths intermediate model

SR/PCEP). These technologies are discussed in the next section.

7.6 Model classification

It is expected that an implementation of the model can be established through a policy framework and supported protocols. But, most importantly, the real life technologies need to be verified with respect to where and how they fit into the model and how they support isolation and encryption with SA dependencies. Hence, a technology classification is needed for each layer.

7.6.1 Network Transport

The network transport layer is the first level of abstraction. A requirement for NFV to work, is that a virtual network between the NFV domains is established. This virtual network consists of NFV Virtual Links [20] that interconnect the NFV domains and create a virtual network (i.e. overlay). The virtual network is created to make end-user traffic traverse through multiple Service Providers' NFV services, without the need of modifying the IP addresses in the original IP packet. The transport technologies reflect the research silos defined in Section 2. Simplified, the cloud computing network providers' offer tunnelling and hardware virtualization(1), the telco research tries to solve this by additional network layers(2), while the datacenter research tries to make networks based on OSI layer two(3).

Cloud based networking: In cloud-based networks, static or dynamic tunnels are set up between the datacenters. Typically Internet Protocol Security (IPSec) or Generic Routing Encapsulation (GRE) tunnels. These tunnels are often set up between the datacenters and additional virtual tunnelling is used inside the tunnels to separate individual service flows. This results in two levels of virtual network transport. The individual channels *must* [21] also be secured and will have a de-

pendency to the underlying channel. This corresponds to the network abstraction model where a higher level of abstractions have SA dependencies upwards in the model. A challenge is that end-to-end encryption for every channel and software based network devices require a lot of computational power, packet overhead and the underlying datapaths are not known.

Additional layers: The key problem is that with VNF forwarding, the IP packets cannot follow the standard routing table. A native IP packet will traverse and potentially flow back and forth between multiple NFV service providers before it reaches the standard Internet routing table. To reduce tunnelling overhead, the telco industry has suggested to solve this without tunnelling, but by using advanced routing. This requires the packet to have additional headers that the routers can base their routing on. One solution is to use additional headers such as Network Service Headers (NSH [26]). Another solution is to use Multiprotocol Label Switching segment routing (MPLS-SR) [27]. (1) NSH natively (without transport tunnels) requires that the routers can read NSH headers. (2) Segment routing has evolved from IETF source routing (SPRING) [27]. It allows the network operator to specify a network path from ingress to egress without using a standard interior gateway routing protocol (MPLS SR/PCEP [28]). This requires interconnected flow control, which the standard Internet Border Gateway Protocol (BGP) tables do not contain today. NSH is the only technology that has suggested an overlay framework for a secure channel between the VNFs [5]. IETF has specified a requirement list for the security extension, but it does not contain any suggestion for vertical security dependency.

Layer two interconnections: Layer two tunnelling (i.e. vlan, 802.1ad, 802.1ah, NV-GRE, STT, VXLAN) [29] has a history of not being sufficiently secure. The protocols do not have any encryption support natively, and need additional tunnelling for that. In a global perspective, most of the protocols have no global address space and do not scale very well. Federated domains are implicitly required. Security considerations such as overwriting MACtables or MAC-to-VTEP mapping, packet insertion and packet sniffing, also makes the protocols vulnerable in respect of end-to-end security.

Discussion: The tunnelling protocol makes interconnections with intermediate Service Providers simple, since the end-to-end tunnelling does not require intermediates to do more than routing the tunnel. The downside is that intermediate providers will not be able to run security features such as mitigating DDoS or offering QoS on different NFV flows and the MTU will in most cases be limited. Therefore, it *can* be concluded that the tunnelling mechanisms are a transit technology until flow based routing, such as segment routing [27] is supported more globally.

7.6.2 Network Control

The Network control layer in the abstraction model refers to the configuration of the network. In its simplest form, it can be a human communication between two network engineers setting up a network between them. The engineers can agree on a "Security Association" and also configure every network flow manually. But, this is not agile and not necessarily fault resistant.

Vertical communication: In NFV Service Chaining, control and configuration of network flows are essential. The ability to signal an NFV traffic path, without using standard destination routing, requires tunnels or flow-based routing.

This paper refers to Software Defined Networking (SDN) as the ability to separate the control plane and data plane traffic. In the abstraction model, the network control layer is responsible for control-plane traffic; the network control layer sends a control signal to the network devices to let it forward packets based on the centralized flow table. This type of signalling also follows three research tracks; datacenters (i.e., OpenFlow [30]), telco (i.e., PCEP [28]) or cloud computing (i.e., OpenvSwitch tunnels [15]).

With centralized control follows the ability to control tunnel state transition, path optimization, repolicying etc. But most importantly, SDN makes the network control accessible through an API in the controller. This makes the network more abstract and programmable with south-, east-west- and northbound interfaces.

Vertical communication from the controller to the network devices affects the end-to-end security. SSL is a standard OpenFlow mechanism to establish these secure network device control channels, but lacks a system for secure key distribution to ensure two-sides authentication. Enhanced options such as the SNBI implementation of OpenDaylight [14] with TPM security can help to ensure two-sides authentication, but only towards the transport layer. No standard for a chain of trust in upwards communication has been defined.

Horizontal communication: Interconnecting two network control domains with respect to SDN does not have many implementations. Primarily there are three ways: (1) making an overlay (FORCES, ALTO, CDNi, I2RS) [30], (2) using an east-west interface (i.e. SDNi[14]) or (3) making a specific protocol (i.e. NSH [26]). East-west interfaces and overlays [14] give a remote peer access to control the whole network. This can be granulated by fine-grained access control, but is difficult to maintain. East-west interfaces also require a distribution of data and states of flows between the operators, that however do not scale. SDN partitioning [31] research has tried to solve the amount of shared resources needed for SDNi [14], but it has resulted in complex access control and policy lists, that are difficult

to operate. Network Service Header signalling has the potential of being scalable, but the instantiation of the tunnel is not specified, other than that it belongs to the network control layer (i.e. OpenFlow, PCRF, Netconf/yang , custom) [3]. This leads back to SDN controller interconnections and leaves a gap in the secure communications on the control-layer.

Discussion: It is debatable whether the interconnection of the control-planes is actually needed and beneficial in the presented abstraction model. It can be claimed that the NFV datapath will be instantiated from the service management level and trigger the network control layer configuration for each domain. This is true if the transport layer tunnels the traffic. If it does not, every forwarding device needs to be aware of the datapath and topology updates. Component topology also sets requirements for secure channel separation for network control. ETSI is open to let the SDN component be run in both a VNF (1), in the infrastructure domain (2) or in the VIM (3). In distributed models such as intermediate or federated models, where network control is outsourced from the datacenter, it is important to differentiate the network control traffic from the data-plane and service control layer. This is to ensure separation of the different organizational roles and different business roles for ISPs.

ETSI has no guidelines for interdomain communication on network control level, but it can be interpreted that they consider this as service-management traffic that belongs to the management layer. This conflicts with their statement about reliability and lack of responsiveness on the management layer [18].

7.6.3 Service Management

The service Management layer is responsible for the lifecycle management of a Virtual Network Function. This is comparable to the Management and Orchestration (MANO) component in the ETSI model. This paper considers all MANO and VNFC-VNFC traffic to be classified as service management traffic.

Interconnecting these components can be implemented by: standardization of protocols (1) or implementation of an APIs such as overlay application or distributed APIs (2). Additionally, the data attributes in the inter-domain communication need a common description. Currently, this communication layer is not standardized and it has no security guidelines. But, multiple research groups work on different models.

Protocol standardization: No standardized network protocol for this layer is found, but IETF has suggested a NSH extension protocol [5]. Dataformat standardization is currently a challenge for designing such protocols. Data format standardization organizations such as: Topology and Orchestration Specification

for Cloud Applications (OASIS TOSCA) and TMforum Information Framework (SID), work specifically with the portability of NFV services and are the leading standard aligned with the ETSI guidelines [32].

Distributed API and overlays: The open-source community of OpenStack has an Infrastructure-as-a-Service approach to the orchestration layer (HOT - Heat Orchestration Templates) [13] and suggests using cascading of the OpenStack API. Extensions and modifications of the OpenStack platform, such as OpenMANO and OPNFV, have also suggested APIs for interconnection. Currently, the open-source environment runs multisite projects [13], but the project has focused on federated models and not models with interconnection between autonomous systems.

Discussion: Recent research [33] shows that interconnections on this layer have functional challenges like lifecycle management, long processing time of both distribution of state information and calculation of Service Graphs.

ETSI started the work on reliability in January 2015 [18] and reliability iscurrently the area of most research. The dependencies between the orchestration components are complex and difficult for operators to manage. This corresponds to the security challenge associated with multiple secure channels between operators. In addition to model security, this paper suggests using the presented security model to also be a basis for function behaviour for interdomain communication interfaces. This would enable aggregation of security policies and automation in the establishment of secure channels.

7.6.4 NFV domain-level

The NFV domain-level in the security model is a new representation of the contractual top-level peering between two ISPs. This paper suggests that; when two ISPs want to use NFV services from each other, they need to set up one top-level interconnection between them. This channel can be used to configure a secure NFV domain relation, and should ultimately contain all data needed to set up the underlying channels. Regarding security, this layer should also be responsible for generating SAs for lower stack layers.

Discussion: There are examples of ISPs [34] that have made custom APIs for their network, allowing customers to integrate with their API and set up network services and peers on demand. These examples could also be considered as "standardization suggestions" on a subordinate level such as the service management level, but are in fact a top-level integration with no service or control channels. The disadvantage is that this makes the application responsible for all network- and service-control. This does not scale for NFV interconnections and makes the API integrations custom and complex.

Class		Attributes					Description	
Abstraction layer	**Sub class**	**Research silo**	**Upper layer support**	**Isolation support**	**Encryption support**	**Security chain support**	**Technologies in class**	**Research status**
Transport	Cloud based networking	Cloud comp.	Yes, but no standard	Yes	Yes, IPsec	Can have	OpenvSwitch+ GRE/IPSEC	Implemented
Transport	Additional Layers	Telco.	Yes, but no standard	Yes	Not yet	Not yet	NSH,MPLS	Implemented
Transport	Layer-2 interconnections	Datacenter network	Yes, industry proprietary	some	No, need tunnels	No	VLAN, VXLAN, STT, NSX, NV-GRE, L2VPN	Implemented
Transport	Plain IP	All	No	some	Yes	No	IP+flowcontrol	Not working
Control	SDN overlay	Cloud comp.	Yes, but no chain	Yes	Not known	not known	FORCES, ALTO, CDNI, I2RS	Implemented
Control	SDN east-west	Datacenter	Yes, but no chain	Yes	Yes, ssl and tunnel	No	OpenFlow, OPFlex ,SDNi	Implemented
Control	Specific Protocol	All	Yes, but no chain	Can have	Can have	No	MPLS SR/PCE, new NSH	Ongoing work
Control	Custom Software-Control	All	Yes, but no chain	Can have	Yes, ssl and tunnel	No	OSVDB, PCRF,BNG, Netconf/Yang	Not known
Control	Abstracted API	All	Not relevant	Can have	Can have	Can have	(OSS/VIM, NorthB-SDN, Custom)	Do not scale
Service	Protocol standarization	Telco.	No	Not yet	Not yet	Not yet	NSH, TOSCA, SID	No specification
Service	Overlay API	Cloud comp.	No	Can have	Yes, ssl and tunnel	No	OpenStackAPI	Implemented
Service	Distributed API	Cloud comp., Datacenter	No	Not yet	Not yet	Not yet	OpenStackAPI, OPNNFV, Open-MANO	Ongoing project
NFV domain	NFV Application	None	Not relevant	Can have	Can have	Can have	Industry proprietary (Colt, NSX)	No specification

Table 7.2: Summary of the NFV technology classification

7.7 Conclusion

Traditionally, end-to-end security is ensured by end-to-end network channels with integrity and encryption. In NFV, there are more than two parties involved in the communication, where the use of traditional end-to-end security methods results in complex setups of network flows. We have created a model that shows there are network security dependencies between the NFV components in a horizontal and vertical manner. The model includes a new top-level integration point on domain-level, that opens up for automation and simplification when deploying NFV interconnections between two ISPs.

The model is also used to classify the research and the NFV interconnection technologies. The technologies have been evaluated with respect to isolation, encryption and ability to communicate up, down sideways in the model. It is shown that the current research silos do not have a common end-to-end security framework and that most technologies lack integrity and encryption. This leaves a security gap in the ETSI NFV model.

From autonomous systems, such as ISPs, NSH is the most promising transport technology that can support the model, but it still lacks support for encryption, integrity and control-layer protocols.

We have introduced a chain of Security Associations (SA) between the NFV components as a possible solution to ensure end-to-end security. Due to lack of dynamics in standard security frameworks, we suggest future work to focus on developing a framework for automatic key distribution of SAs such as block-chains or key tokens [35].

References

[1] ETSI. *Network Function Virtualization (NFV) Use Cases 001 v1.1.1*. Available online: http://www.etsi.org/deliver/etsi-gs/nfv/001_099/001/01.01.01-60/gs-nfv001v010101p.pdf (accessed on 04 June 2019). 2013.

[2] ETSI. *Network Functions Virtualisation (NFV) NFV-SEC 001 Problem Statement*. Available online: http://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/001/01.01.01_60/gs_NFV-SEC001v010101p.pdf (accessed on 04 June 2019). 2014.

[3] Rashid Mijumbi et al. 'Network Function Virtualization: State-of-the-art and research challenges'. In: *IEEE Communications Surveys & Tutorials* 18.1 (2015), pp. 236–262.

[4] 5GPPP Architecture WG. *View on 5G Architecture*. Available online: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-View-on-5G-Architecture-For-public-consultation.pdf (accessed on 04 Aug 2016). 2016.

[5] K Tirumaleswar Reddy et al. *Authenticated and encrypted NSH service chains*. Internet-Draft. Work in Progress. Apr. 2015.

[6] Norival Figueira et al. *Policy Architecture and Framework for NFV Infrastructures*. Internet-Draft draft-irtf-nfvrg-nfv-policy-arch-04. Work in Progress. Internet Engineering Task Force, Sept. 2016.

[7] Open Networking Foundation (ONF). *Functional Requirements for Transport API (TR-527)*. Available online: https://www.opennetworking.org/wp-content/uploads/2014/10/TR-527_TAPI_Functional_Requirements.pdf (accessed on 04 June 2019). 2016.

[8] Max Alaluna, Fernando MV Ramos and Nuno Neves. 'Literally above the clouds: Virtualizing the network over multiple clouds'. In: *IEEE NetSoft Conference and Workshops (NetSoft)*. IEEE. 2016, pp. 112–115.

[9] Roberto Bifulco, Anton Matsiuk and Alessio Silvestro. 'Ready-to-deploy service function chaining for mobile networks'. In: *IEEE NetSoft Conference and Workshops (NetSoft)*. IEEE. 2016, pp. 175–183.

[10] Marc G Villinger and Reinhard Jung. 'Establishing a continuous corporate business model innovation process: Process antecedents'. In: (2015).

[11] Bram Naudts et al. 'Deploying SDN and NFV at the speed of innovation: Toward a new bond between standards development organizations, industry fora, and open-source software projects'. In: *IEEE Communications Magazine* 54.3 (2016), pp. 46–53.

[12] Ahmed Abujoda and Panagiotis Papadimitriou. 'DistNSE: Distributed network service embedding across multiple providers'. In: *8th International Conference on Communication Systems and Networks (COMSNETS)*. IEEE. 2016, pp. 1–8.

[13] OpenStack community. *The OpenStack API webpage*. Available online: http://docs.openstack.org/developer/networking-sfc/api.html (accessed on 01 August 2016). 2016.

[14] Hongtao Yin et al. *SDNi: A Message Exchange Protocol for Software Defined Networks (SDNS) across Multiple Domains*. Internet-Draft draft-yin-sdn-sdni-00. Work in Progress. Internet Engineering Task Force, June 2012.

[15] OPNFV and OpenvSwitch community. *The Open Virtual Network*. Available online: https://wiki.opnfv.org/display/PROJ/Ovn4nfv (accessed on 01 August 2016). 2016.

[16] ETSI. *Network Functions Virtualisation (NFV) NFV-SEC 003 Security and Trust Guidance*. Available online: http://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/003/01.01.01_60/gs_NFV-SEC003v010101p.pdf (accessed on 04 June 2019). 2014.

[17] ETSI. *Network Functions Virtualisation (NFV) 002 Architectural Framework v1.1.1.* Available online: http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_nfv002v010101p.pdf (accessed on 04 June 2019). 2014.

[18] ETSI. *Network Functions Virtualisation (NFV) NFV-REL 003 Models for End-to-End Reliability*. Available online: http://www.etsi.org/deliver/etsi_gs/NFV-REL/001_099/003/01.01.02_60/gs_nfv-rel003v010102p.pdf (accessed on 04 June 2019). 2016.

[19] ETSI. *Network Functions Virtualisation (NFV) NFV-EVE 005 SDN Usage in NFV Architectural Framework*. Available online: http://www.etsi.org /deliver/etsi_gs/NFV-EVE/001_099/005/01.01.01_60 /gs_nfv-eve005v010101p.pdf (accessed on 04 June 2019). 2015.

[20] ETSI. *Network Functions Virtualisation (NFV) NFV-MAN 001 Management and Orchestration*. Available online: http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/ gs_nfv-man001v010101p.pdf (accessed on 04 June 2019). 2014.

[21] Scott O. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. RFC 2119. Mar. 1997.

[22] A Jøsang. 'Prospectives for modelling trust in information security'. In: *Australasian Conference on Information Security and Privacy*. Springer. 1997, pp. 2–13.

[23] Randall Atkinson and Stephen Kent. *Security Architecture for the Internet Protocol*. RFC 2401. Nov. 1998.

[24] Network Working Group et al. 'Internet X. 509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile'. In: *RFC5280* (2008).

[25] Abhishek Gupta et al. 'Joint virtual network function placement and routing of traffic in operator networks'. In: *University of California Davis, USA, Tech. Rep* (2015).

[26] Paul Quinn and Uri Elzur. *Network Service Header*. Internet-Draft draft-ietf-sfc-nsh-05. Work in Progress. Internet Engineering Task Force, 2016.

[27] Clarence Filsfils et al. 'The Segment Routing Architecture'. In: *2015 IEEE Global Communications Conference (Globecom)*. IEEE. 2015, pp. 1–6.

[28] JP Vasseur, Adrian Farrel and Gerald Ash. *A Path Computation Element (PCE)-Based Architecture*. RFC 4655. Aug. 2006.

[29] Siamak Azodolmolky, Philipp Wieder and Ramin Yahyapour. 'SDN-based cloud computing networking'. In: *15th International Conference on Transparent Optical Networks (ICTON)*. IEEE. 2013, pp. 1–4.

[30] Sandra Scott-Hayward, Sriram Natarajan and Sakir Sezer. 'A survey of security in software defined networks'. In: *IEEE Communications Surveys & Tutorials* 18.1 (2015), pp. 623–654.

[31] Teemu Koponen et al. 'Network virtualization in multi-tenant datacenters'. In: *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*. 2014, pp. 203–216.

[32] Hassan Hawilo et al. 'NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC)'. In: *IEEE Network* 28.6 (2014), pp. 18–26.

[33] J Garay et al. 'Service description in the NFV revolution: Trends, challenges and a way forward'. In: *IEEE Communications Magazine* 54.3 (2016), pp. 68–74.

[34] Colt. *Case Study Rolling out hybrid cloud services across Europe from Juniper.* Available online: http://www.colt.net/blog/2014/05/23/sdn-nfv-the-beginning-of-a-new-era (accessed on 01 August 2016). 2015.

[35] D.W. Bachmann et al. *Token caching in trust chain processing*. US Patent 9,325,695. Apr. 2016.

Chapter 8

Security Requirements for Service Function Chaining Isolation and Encryption

Published in IEEE 17th International Conference on Communication Technology (ICCT), Chengdu, China, 2017

Håkon Gunleifsen, Thomas Kemmerich

Faculty of Information Technology and Electrical Engineering
Norwegian University of Science and Technology, Gjøvik, Norway

Email: hakon.gunleifsen2@ntnu.no thomas.kemmerich@ntnu.no

Abstract

This paper presents a study of Service Function Chaining (SFC) isolation and encryption in interconnected Network Function Virtualisation (NFV) domains. The adoption of NFV deployments is currently designed to be implemented within trusted domains where overlay networks with statically trusted links are considered to enable network security. We challenge this statement and introduce a security problem related to Virtual Network Functions (VNF) confidentiality and isolation. A data-flow that traverses through a chain of Virtual Network Functions (VNF) cannot be end-to-end encrypted when each VNF must have access to the data-flow. This restricts both end-users and Service Providers from enabling end-to-end security and VNF isolation to their NFV flow. Therefore, there is a need to encrypt the data-flows on a per flow basis. In this paper we present the discovered security problem, set the requirements for the problem solu-

tion and study the constraints for securing and isolating VNFs in a Service Function Chain.
Keywords: Network Function Virtualisation, Service Function Chaining, Network Service Headers, Network Encryption, NFV

8.1 Introduction

In a Service Function Chain (SFC) [1], all Virtual Network Functions (VNFs) are designed to have access to the content of a datastream. We challenge this concept and state that VNFs should not have access all the data in an SFC. To prevent data from being eavesdropped, an end-user requires an SFC with the capability of isolating the different Virtual Network Functions from each other. We also define that the end-user requires that the traffic is encrypted for each Virtual Link [1] in the SFC. The SFC specification does not allow such isolation. Neither is encryption of Virtual Links supported without additional transport protocols. To allow such functionality, we define a security problem and study the functional security requirements to support such a model.

This paper is organised into four parts. First, the security problem and the related work is presented (Section: 8.1,8.2). Second, the requirements for such a model are presented (Section: 8.3). The constraints and solution possibilities with the existing technologies and specifications are presented in Section 8.4. Finally, we conclude the paper (Section: 8.5).

8.1.1 The Security Problem

The figure (Fig.8.1) shows a classical inter-domain SFC example in NFV, where third-party VNFs are allowed to intercept and modify the user traffic.
The end-user has agreed to allow the NFV providers to run a set of Virtual Functions. Provider A establishes an SFC and classifies i.e. VOIP and HTTP traffic to follow one SFC path. The user wants to make sure that the VNFs only have the right to modify their respective parts of the data traffic. Hence, we aim to ensure that the data traffic is secured from eavesdropping from all unauthorised parties. For example, VNF A can access the HTTP traffic and not VOIP traffic, while VNF B can access VOIP traffic, but not HTTP. VNF C has access to both HTTP and VOIP. Currently, this is not possible and every VNF has access to all services defined by the SFC.

Because destination based IP routing does not work for NFV service chain routing, the standard encryption methods of the Internet Protocol (IP) are no longer easily applicable. The main problem is that VNFs are not the destination of the IP packet, but they are middleboxes the packets must traverse. In fact, an IP packet

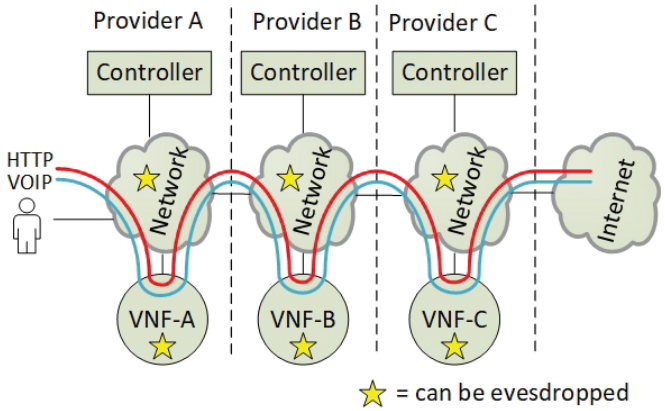


Figure 8.1: The eavesdropping problem in Service Function Chaining

can potentially traverse unmodified through the SFC. To ensure that the IP packets traverse all the VNFs, the network must support flow-based routing that routes IP packets based on more attributes than only destination IP addresses. To ensure per-user and per-flow encryption, each data-flow has to be individually and differently encrypted based on a set of flow classification attributes such as destination IPaddr, source IPaddr, port, etc. These are the same attributes used for NFV classification [1]. However, if the packets are encrypted, the NFV classifiers are no longer capable of classifying or routing the packets according to the flow specification. Hence, the route path of the packets must be determined before the packets are encrypted. This makes the first hop in the service chain, the classification service, responsible for the encryption and the routing mechanism of the encrypted packets. The SFC architecture has no solution to enable such encryption, and that implies that there is a security gap in the architecture.

The need for additional encryption services also includes a need for automation for secure channel configuration. The number of encrypted channels grows exponential with respect to the number of VNFs and VNF traffic access levels. Therefore, automation in key creation is needed to support a larger scale of secure channels. To enable automation, the encryption channel must also be identified. The SFC specification [1] does not contain any method to verify if a Virtual Link is encrypted. It has no encryption identifier for the different flows and it has no architecture for key agreement automation.

Hence, an architectural challenge and a gap in the SFC model is discovered with respect to 1) VNF isolation in a single SFC and 2) SFC encryption automation and validation. This paper suggests a set of requirements to the problem solution and

discusses the constraints of existing technologies.

8.2 Related Work

There is currently no specification of how to enable isolation and encryption of SFC traffic in NFV. The SFC specification [1] only states that encryption can be ensured by the NFV transport protocol. The transport protocol layer is not the OSI model layer 4, but the outer transport layer that transports a data-packet between two VNFs. This layer often corresponds to the Virtual Link channel between two VNFs. The transport layer can enable encryption, but the consequences of applying encryption depend on the transport protocols' encryption capability. In most cases, applying encryption to an IP packet results in a lack of readable headers, which prevents routers from routing packets correctly through an SFC (see Section: 8.4.3). Hence, the related work primarily concerns the transport protocols and their capability to do flow-based routing in combination with encryption. There are currently three main methods to route SFCs. 1: The flow-based approach (OpenFlow/BGP FlowSpec) [2], 2: Additional header encapsulation (NSH,MPLS/SR) [3], [4], 3: Tunnelled overlay networks (IPsec/l2TP) [5].

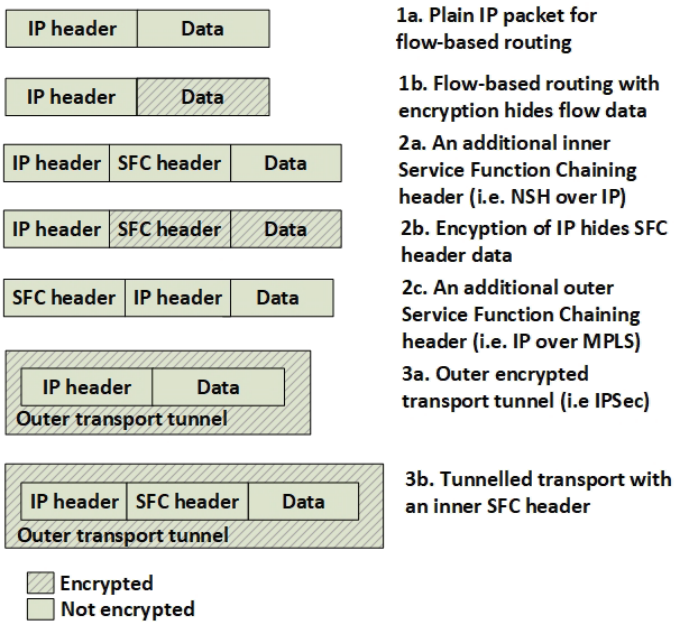


Figure 8.2: Methods to encapsulate and route NFV traffic

1) The flow-based approach: Flow-based networks can enable simple SFC routing by reading packet information from a plain IP packet (Fig.8.2.1a). A flow-based routing approach, commonly used in for example OpenFlow networks, is to enable a global address space for flow-based routing and distribute the flow specific routes across all NFV Providers. This can be enabled with a common protocol such as FlowSpec BGP [2]. It requires that flow-specific attributes are exposed in the packets and that they are uniquely identifiable for routers across multiple domains. Encrypting packets will hide attributes such as UDP/TCP ports and it makes flow-based routing impossible (Fig.8.2.1b). However, in some cases, such as IPSec ESP transport mode, OpenFlow can make route decisions of encrypted traffic [6]. For advanced SFCs, such as multiple occurrences of the same VNF in an SFC, there is not even enough information in a plain unencrypted IP data packet to make precise routing decisions. Therefore, additional packet header encapsulations is specified by the Internet Engineering Task Force (IETF).

2) Additional header encapsulation: By putting on an additional header to layer 2, 3 or 4 packets, the routers can perform a new form of flow-based routing based on the information found in an additional inner or outer packet header. The most common additional headers are Network Service Headers (NSH) [3] and Multi Protocol Label Switching (MPLS) label headers [4]. Common for both these technologies is that all the related routers must support routing based on these headers. Because not all routers support these different headers, encrypted outer overlay tunnels can be established between routers that support one of these technologies. The additional headers themselves do not enable any encryption features, but they only ensure SFC routing.

Network service headers (NSH) is a technology that supports routing and route path integrity of SFC. The IETF NSH working group has specified an integrity check mechanism [7] to ensure that NSH headers cannot be modified. This NSH header verification is limited to NSH header integrity check only and not to data encryption. The NSH working group suggests to use an outer transport network encryption with an NSH integrity check to enable security. If the transport of NSH is encrypted and the inner NSH header is verified as correct, then NSH can be perceived as a technology that supports hop-by-hop encryption in SFCs (Fig.8.2.3b). However, when NSH is put on top of IP without an outer encrypted transport protocol, encrypting the IP packet also includes encrypting the NSH header. That implicitly hides the additional header for the routers (Fig.8.2.2a/2b). The only valid alternative is therefore to use an overlay transport network to setup encrypted links between every NSH-aware [1] router. Such additional overlay tunnels only enable hop-by-hop tunnels that are not known to end-users. Hence, end-to-end encryption is not supported.

IETF has also suggested solving the SFC in NFV by using MPLS label headers. When utilising segment routing for MPLS (MPLS/SR) [4], MPLS can encapsulate an encrypted packet with a chain of MPLS labels (Fig.8.2.2c). Since this is an outer transport header, the inner content of the MPLS packet can stay encrypted without loosing the routing information. That makes it possible to route an encrypted packet according to the SFC path. Unfortunately, when the packet leaves the MPLS domain to another service provider and needs reclassification, the routing information can get lost if the packet is encrypted. Seamless MPLS tries to solve that problem, but it requires a uniform MPLS network with inter-domain agreements of routing, that does not easily support a global SFC architecture.

Segment routing for IPv6 [8] gives encryption support in the IPv6 protocol itself. However, during encryption, the header is replaced and the segments are lost.

3) Tunnelled overlay networks: A tunnelled overlay network is typically archived by setting up IPsec tunnels [5] between every instance of VNFs or between NFV data-center routers (Fig.8.2.3a). Without an additional routing mechanism such as flow-based routing or the use of additional packet headers, there is currently no common policy method to direct service chained traffic in and out of such overlay tunnels. Neither can multiple encrypted flows be differentiated from each controller. Hence, tunnelled overlay network must be used in combination with a flow-based routing mechanism.

8.3 Requirements

The solution to the problem is clearly to enforce encryption and to extend the number of encrypted channels over the Virtual Links. Enabling such functionality does not fit the current SFC architecture and it sets additional requirements for new components to support instantiation of new encryption tunnels. This section discusses these requirements to fully enable VNF isolation with SFC traffic encryption.

8.3.1 Dynamic Tunnels

Currently, cross-domain encrypted SFC transport links between NFV Service Provider's data-centers are primarily set up manually. However, when the number of domain participants, VNF services and links grows, it does not scale to set up encrypted tunnels manually. Multiple secure tunnels between service providers, such as one secure channel per Virtual Link, dramatically increase the number of tunnels to maintain. That does not scale with respect to changing keys and setting up and removing secure tunnels. The setup of secure channels must be automated in such a way that manual input is not needed. This means that we need an open distributed system with the ability to authenticate both ends of the secure channels. An additional trusted control channel is required to instantiate such dynamic

secure tunnels.

8.3.2 New Identifiers

There is a need of new identities to be able to authenticate each Service Provider and their NFV services. The identities must be individually connected to a set of encryption keys. The identifiers are associated with an IP flow, an NFV service ID, Virtual Links and an NFV Service Provider ID. This means that endpoints of the secure channels have different characteristics with different identifiers than Virtual Links. These encryption identifiers need to be defined.

8.3.3 Service Function Chaining Integrity

The problem description (Fig.8.1) shows that end-to-end security is needed in SFC. In NFV service chains, the middleboxes are authorised to have access and alter some parts of the user data, that compromise the end-to-end integrity and confidentiality paradigm. Nevertheless, the middleboxes are the only actors that are authorised to modify the data. The end-user wants to ensure that the authorised middlebox is the only party that has this access. Therefore, the integrity must be secured in such a way that the end user can specify that the involved VNFs can only read and modify what they are allowed to do. Specifically, what IP flows the VNFs can access and the end users ability to get information about how the Virtual Links are encrypted. Additionally, it must be ensured that the flow-specific routes, the SFC paths and the SFC packet headers cannot be modified by unauthorised parties.

8.3.4 Flexibility in Encryption Types

Every Service Provider that offers NFV services must announce their service capabilities to other Service Providers to be able to automate encryption setup. Encryption algorithms and protocols continue to evolve. Hence, the NFV services must support a set of different standards that easily can be extended. It is expected that multiple encryption methods must be supported by different SCFs, but also different encryption per hop should be possible. Therefore the encryption capabilities for each Virtual Link must be agreed upon during SFC setup and not strictly be limited to one encryption method.

8.3.5 A New East-West Communication Channel

When Internet Service Providers interconnect to utilise each others NFV services, their security policy often implies that the SDN controllers cannot be federated for flow-table sharing. The SDNi protocol [9] is designed to control multiple SDN domains of one operator only. This means that the Virtual Network Function Components (VNFCs) cannot share one global network controller across NFV operator

domains. When the VNF Forwarding Graph [1] cannot be controlled globally, there is a need for a new protocol to interconnect the network controllers. Currently, non-federated SDN controllers can only communicate east-west by BGP-related protocols such as PCEP [10] or BGP FlowSpec [2]. These protocols do not contain any information relevant to the setup of dynamic encryption channels with SFC routing. Hence, in addition to for example BGP FlowSpec, a new east-west communication channel is needed to exchange SFC route attributes.

8.3.6 Multi-protocol Support

To support multiple types of transports and encryption methods, there must be a configurable choice in the setup of the secure channels and the key agreements. First, the transport mechanism of the service chain must be agreed upon, then the protocol-specific encryption related to the transport can be exchanged. This multilevel setup of Virtual Link transport and encryption tunnels must be a part of the new east-west communication channel between the network controllers in the different domains.

8.3.7 Key Management Service

Key Distribution Services (KMSs) are needed to distribute the keys of the encryption protocol setup to all encrypted links in an SFC. KMSs have the advantages that they support a large number of peers and open up for a more flexible and scalable encryption setup. On the other hand, a KMS such as Kerberos [11], is based on a single server and a single root of trust. The architecture must support multiple KMS servers that all can be trusted. That requires a secure distribution of trusted KMSs between the different network domains.

8.3.8 Hop by hop encryption

The VNF isolation problem (see Section: 8.1.1) indicates that data encryption is needed between the different VNFs. To allow backwards compatibility with standard encryption protocols, a model must support pairwise hop-by-hop encryption of specific flows. This implies that every Virtual Link must be split into multiple encryption channels in a flow-based manner. One option to enable VNF isolation in a standard SFC architecture [1] is to branch every hop in the SFC (Fig.8.3.2). Branching is allowed according to the SFC specification [1], but branching every VNF is not preferred since it breaks the whole concept of SFC. If branching was used to solve the isolation problem, then merging would also be required. Merging branches should be avoided according to the specification [1]. Therefore, SFC branching cannot be used to support flow-based encryption. Hence, additional flow-based encryption tunnels inside the Virtual Links are required to archive isolation. To maintain the SFC and enable encryption of specific flows, the encrypted

tunnels must also be capable of bypassing the VNFs that are not allowed to modify the data flow (Fig.8.3.3). To be able to support such an architecture, the SFC spe-

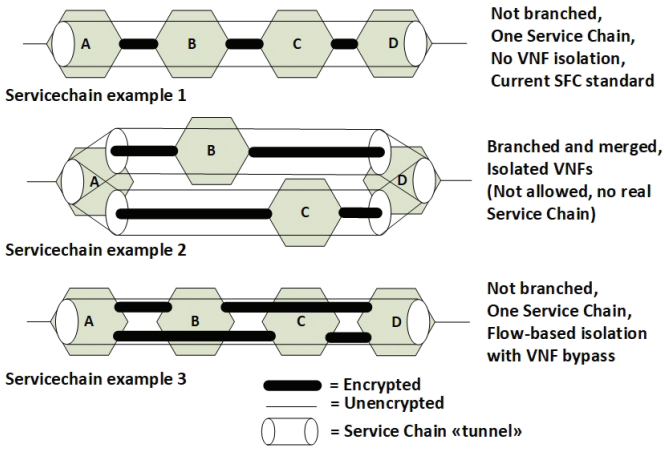


Figure 8.3: Examples of Service Function Chaining with encryption

cification has to include flow specific rules that specify what VNFs to bypass. The bypass traffic is in this setup encrypted. To be able to classify encrypted bypass traffic, the encrypted packets must contain enough readable packet information to make them classifiable. That means that the SFC packet headers themselves cannot be encrypted because it prevents VNF bypassing and flow-based routing.

8.3.9 Reliance

The routes in the underlying network may change dynamically, but the SFC path cannot be dependent on network topology changes. This also implies that the encryption keys cannot change for every route change. On the other hand, VNFs may become unavailable and the SFC is required to change its path. That means that encryption keys must also change. It is important that an encryption architecture supports such topology changes. To minimize packet loss during network or VNF application failure, backup encryption tunnels should be preconfigured to ensure fast reroutes. The control plane is not responsible for defining the SFC, but it receives the SFC route information from the service plane. To allow fast reroutes, the control plane should also be populated with SFC backup paths. Then, the control plane can be allowed to make SFC path alterations during network failures. Making backup paths of all possible combinations of VNF failures creates an exponential set of extra tunnels and a lot of computational overhead. However, a set of allowed bypass backup tunnels defined in the service plane, is an opportunity that can increase the resilience of the SFCs.

8.4 Architectural Constraints and Opportunities

This section discusses the most important constraints and possibilities related to the defined requirements.

8.4.1 Encryption as a VNF Attribute

The SFC specification states that the network routing and the SFC should not be mixed with the VNF application [1]. That means that the encryption service cannot be an attribute of the VNF itself. This is because one VNF can occur multiple times in an SFC and therefore it may have multiple Virtual Links and encryption interfaces. This implies that the encryption of data traffic inside the VNF is against the specification and limits the use of encryption software inside the VNF application.

8.4.2 Double Encryption Avoidance

If the Virtual Link transport protocol enables encryption, such as an IPsec channel, then encryption is supported by the Virtual Link itself. If no isolation is needed, then there is no need for additional encryption. It is assumed that "no need" for isolation is the most common end-user requirement. Therefore, encryption per Virtual Link can in most cases be enough to protect SFCs from eavesdropping. If the Virtual Link is encrypted either by an outer transport protocol or by the protocol itself, that information should be announced to the network controllers responsible for the encryption. To avoid additional and unnecessary encryption, this information is an important attribute to the Virtual Link. Hence, the characteristics of this Virtual Link transport and the quality of the encryption level can be announced between the network controllers. Virtual Link protocols such as MPLS/SR and NSH do not support encryption by themselves. However, NSH is able to perform header integrity checks [7] that raise the security level of the Virtual Link transmission. This is in fact also a security attribute to the Virtual Link and the SFC, which can also be announced between the controllers. Knowing this information gives possibility to allow the Service Providers and end-users to set requirements to their SFC.

8.4.3 Header Visibility

The requirement section stated that encrypted traffic over a Virtual Link must contain readable packet classification information (see Section: 8.3.8). Encrypting a packet over a Virtual Link will hide OSI layer 3 to 7 packet information, that is important for SFC classification and routing. Therefore, the packets must be classified before they enter the encryption service. To preserve the classification information after encryption, the classification data needs to be preserved as readable in

the packets even after encryption. That means that encrypted packets must be extended with additional readable SFC headers. This fact constrains the SFC routing solution outcomes down to the use of MPLS/SR and NSH headers only. However, encrypting packets with such additional SFC headers also has constraints. Service Function Chaining with the use additional SFC headers can be utilized by putting on the additional header outer or inner to the IP layer. SFC headers inside IP (i.e. NSH over IP) require a new SFC-aware encryption protocol that does not encrypt the SFC header. Such an architecture requires extensions to the classical IPSec specification, where encrypted packets must contain more readable packet information next to the IP header, such as readable SFC headers inside an IPsec packet. This is not a problem when the SFC headers are put outside the IP header such as MPLS. However, headers outer to IP requires that the headers must be visible for everyone contributing to an SFC. When a packet leaves an MPLS domain, this header information can be lost. This implies a practical constraint, because it enforces a global routing mechanism for outer headers such as a global MPLS network.

8.4.4 Computational power

The dynamic tunnel setup and the flow encryption services increase the need of computational power. The massive amount of encrypted channels can both slow down the link performance and overuse data-center resources. It is possible to offload the service plane for encryption processes by utilising data plane resources, but this does not reduce the total resource consumption in a data-center. However, a control plane instantiated encryption service is closer to hardware, which gives an opportunity for hardware encryption assistance. It is expected that a programmable data plane [12] becomes more available on enterprise switches and routers. When this hardware is combined with control plane flow control, hardware accelerated encryption is considered feasible.

8.4.5 MTU Increasing

Encrypted data packets often include an increase in the Maximum Transfer Unit (MTU) and implicitly also resulting packet segmentation. This is considered normal in any encryption setup. However, an introduction of multiple additional encryption layers on the data plane with resulting packet segmentation, can result in lower performance of packet forwarding.

8.5 Conclusion

In this paper, we presented a security gap in the NFV architecture related to Virtual Link isolation and encryption in inter-domain topologies. That resulted in a set of requirements to a problem solution. We also elaborated about the architectural

constraints related to existing standards and technologies. We conclude that there is a security gap in the SFC specification and that a set of new requirements has to be a part of this specification.
We aim to continue our work with the solution to the problem by focusing on a solution architecture with automated tunnel setup and cross-domain Key Management Services.

References

[1] Joel M. Halpern and Carlos Pignataro. *Service Function Chaining (SFC) Architecture*. RFC 7665. Oct. 2015.

[2] Pedro R. Marques et al. *Dissemination of Flow Specification Rules*. RFC 5575. Aug. 2009.

[3] Paul Quinn, Uri Elzur and Carlos Pignataro. *Network Service Header (NSH)*. Internet-Draft draft-ietf-sfc-nsh-19. Internet Engineering Task Force, Aug. 2017. 34 pp.

[4] C. Filsfils et al. *Segment Routing with MPLS data plane*. Internet-Draft draft-ietf-spring-segment-routing-mpls-10. Internet Engineering Task Force, June 2017. 11 pp.

[5] Randall Atkinson. *Security Architecture for the Internet Protocol*. RFC 1825. Aug. 1995.

[6] Vahid Heydari Fami Tafreshi et al. 'Integrating IPsec within OpenFlow architecture for secure group communication'. In: *ZTE Communication Journal* 12.2 (2014), pp. 41–49.

[7] K Tirumaleswar Reddy et al. *Authenticated and encrypted NSH service chains*. Internet-Draft. Work in Progress. Apr. 2015.

[8] Clarence Filsfils et al. 'The Segment Routing Architecture'. In: *2015 IEEE Global Communications Conference (Globecom)*. IEEE. 2015, pp. 1–6.

[9] Hongtao Yin et al. *SDNi: A Message Exchange Protocol for Software Defined Networks (SDNS) across Multiple Domains*. Internet-Draft draft-yin-sdn-sdni-00. Work in Progress. Internet Engineering Task Force, June 2012.

[10] JP Vasseur and Jean-Louis Le Roux. *Path Computation Element (PCE) Communication Protocol (PCEP)*. RFC 5440. Mar. 2009.

[11] Kenneth Raeburn. *Encryption and Checksum Specifications for Kerberos 5*. RFC 3961. Feb. 2005.

[12] D. Perino et al. ‘A programmable data plane for heterogeneous NFV platforms’. In: *IEEE Conference on Computer Communications Workshops (Infocom Workshops)*. Apr. 2016, pp. 77–82.

Chapter 9

A Tiered Control Plane Model for Service Function Chaining Isolation

Published in Multidisciplinary Digital Publishing Institute (MDPI)
Journal; Future Internet, 2018

Håkon Gunleifsen *, Vasileios Gkioulos and Thomas Kemmerich

Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology, Postbox 191, 2802 Gjøvik, Norway;

hakon.gunleifsen2@ntnu.no, vasileios.gkioulos@ntnu.no, thomas.kemmerich@ntnu.no

Abstract

This article presents an architecture for encryption automation in interconnected Network Function Virtualization (NFV) domains. Current NFV implementations are designed for deployment within trusted domains, where overlay networks with static trusted links are utilized for enabling network security. Nevertheless, within a Service Function Chain (SFC), Virtual Network Function (VNF) flows cannot be isolated and end-to-end encrypted because each VNF requires direct access to the overall SFC data-flow. This restricts both end-users and Service Providers from enabling end-to-end security, and in extended VNF isolation within the SFC data traffic. Encrypting data flows on a per-flow basis results in an extensive amount of secure tunnels, which cannot scale efficiently in manual configurations.

Additionally, creating secure data plane tunnels between NFV providers requires secure exchange of key parameters, and the establishment of an east–west control plane protocol. In this article, we present an architecture focusing on these two problems, investigating how overlay networks can be created, isolated, and secured dynamically. Accordingly, we propose an architecture for automated establishment of encrypted tunnels in NFV, which introduces a novel, tiered east–west communication channel between network controllers in a multi-domain environment.

Keywords: software defined networks; service function chain; virtual network functions; border gateway protocol; traffic isolation; key management services

9.1 Introduction

This article builds on the need for end-to-end encryption and traffic isolation between services in Network Function Virtualization (NFV) with Service Function Chaining (SFC), for which no automation method or standardization currently exists. In a chain of multiple NFV services, the intermediate Virtual Network Functions (VNFs) aka middle-boxes require access to the content of the data-stream, which makes end-to-end encryption impossible. In a simplified example of similar nature, a caching HTTP proxy server must have access to the HTTP content in order to be able to cache. Therefore, in order to enable encrypted SFCs, the middle-boxes must take part in the encryption services. Enabling secure channels in such setups relies on establishing hop-by-hop secure channels, which are collectively perceived as end-to-end secure.

We argue that, by adopting this encryption principle in SFC, a sum of hop-by-hop secure channels can enable end-to-end security. Furthermore, we argue that, in a dynamic NFV environment where the VNF can move between data-centers and change order in an SFC, the establishment of secure channels must be automated and the integrity of each hop must be verified dynamically, in order to allow for scalability and dynamic adaptation. Accordingly, in this article, we propose a new architecture with an additional data packet header with a corresponding new east–west communication protocol on the control plane (Figure 9.1). The architecture is presented in a top–down approach, focusing and discussing at four primitive levels of abstraction for completeness, namely: 1—Model, 2—Service, 3—Protocol and interface, 4—Implementation.

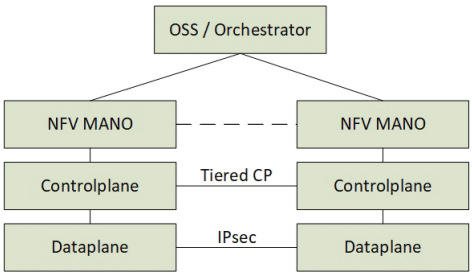


Figure 9.1: Extended east–west communication for Network Function Virtualization.

This article is organized as follows: the Introduction section is followed by an overview of related work in Section 9.2. Section 9.3 presents the top-level architecture and model of our contribution. The functionality of the components in the architecture is presented in Section 9.4, while Section 9.5 explains the communication between the services. The Implementation Guidelines in Section 9.6 give a short overview of how a subset of the most important components can be implemented and a simulated proof of concept. Section 9.7 simulates a proof of concept implementation and discusses the limitations of the architecture. Sections 9.8 and 9.9 suggest future work and conclude this article.

9.1.1 Research Challenges

Figure 9.2 shows where end-to-end user traffic can be eavesdropped on in an SFC. NFV is designed to be flexible and simple where most intermediate NFV services are perceived as transparent network services. Hence, the common end-users are unaware of the potential of traffic eavesdropping in an SFC. In multi-operator and multi-tenancy NFV networks, we argue that end-users need to know which operators require access to their traffic. In addition, we assume that the end-users want to apply security policies to their SFCs, which would specify what type of SFC traffic each operator can access. We argue that the end-users require the flexibility of both allowing a subset of the VNFs to have access to all their data traffic, and letting other subsets of the VNFs have access only to a specific type of traffic. Currently, such traffic isolation within an SFC is not possible; end-to-end encryption is also not supported within an SFC.

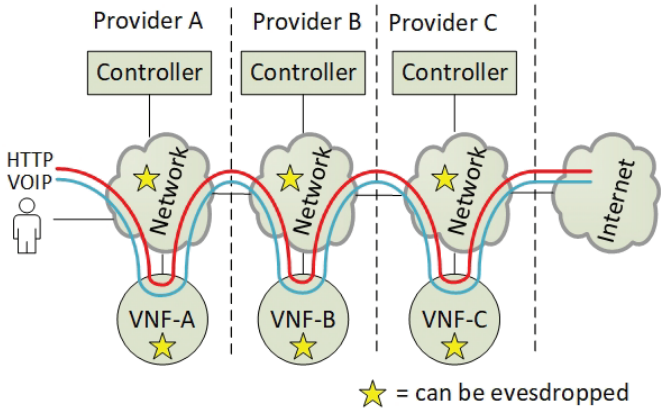


Figure 9.2: The adversary model.

This article introduces an architecture that isolates and encrypts SFC traffic between the different VNFs, which requires the automation of the encryption setup. Enforcing the network traffic through encryption services requires a new east–west protocol for network routing and for key derivation. Hence, this article also suggests a protocol and a procedure that can automate this dynamic routing and encryption setup.

Figure 9.3 shows different approaches in routing SFC traffic. This is mainly reflected by plain flow-based routing (Figure 9.3(1,2)), the use of additional SFC headers (Figure 9.3(3,4,5)) or by using transport tunnels (Figure 9.3(6,7)). Running common encryption services on the Internet Protocol (IP) layer introduces an SFC routing problem because the packet encryption hides or changes the meta-data information such as the destination port inside the IP packet (Figure 9.3(2)). This lack of meta-data makes flow-based routing with i.e., OpenFlow difficult without using additional packet header tagging such as Network Service Headers (NSH) [1] or Multi-protocol Label Switching (MPLS) [2]. Hence, the packets must be classified before they are encrypted and a packet tag must be applied to the IP packet in order be able to route the packet correctly. Our solution to this problem is to put an SFC header in front of the encrypted packet and encapsulate it by a transport tunnel running between the NFV providers (Figure 9.3(7)).

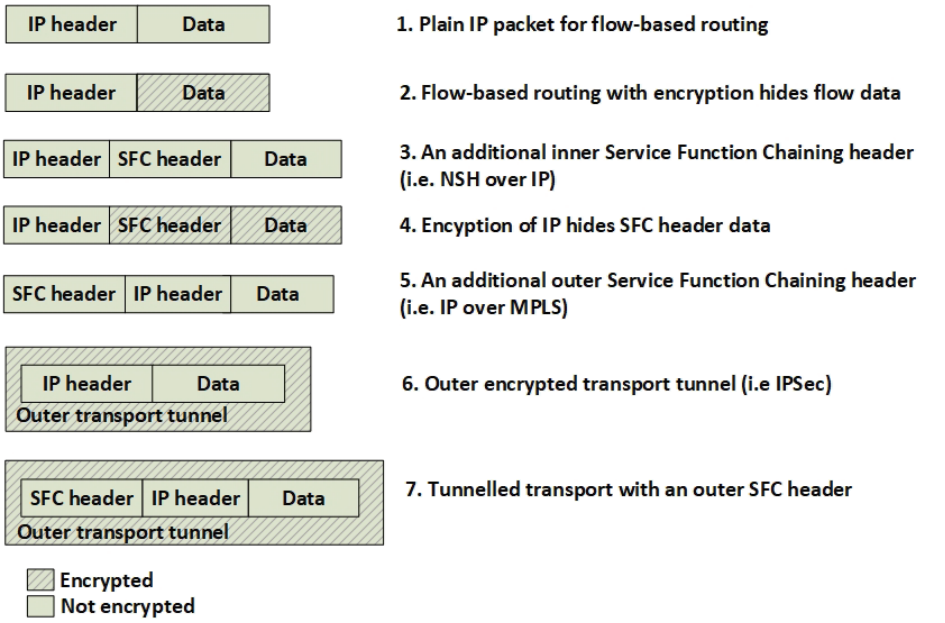


Figure 9.3: Encryption possibilities.

Another related problem when not using SFC headers such as NSH or MPLS is how to ensure the state of an SFC packet. Figure 9.4 shows that an SFC can traverse back and forth between two NFV Service Providers. The example shows how the network device B must know if an incoming packet from the tunnel has its destination to VNF 2 or VNF 4. Since the IP headers of the SFC packet normally do not change, the packet must be tagged by meta-data or be tunnelled.

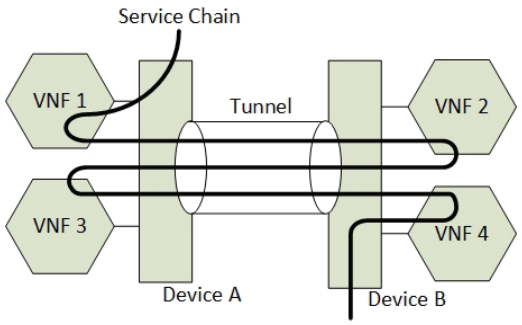


Figure 9.4: Flow identification problem.

This need for SFC packet headers excludes most SFC technologies that are not using additional SFC headers from being applied in combination with IP encryp-

tion. This also excludes other methods that do not allow the use of NSH or MPLS. Hence, this article focus on using NSH as both an SFC packet forwarding and a routing mechanism.

9.2 Related Work

Both our previous work [3] and other related research [4] have identified many technologies in order to interconnect NFV services across multiple service providers. The research shows that, due to a wide set of components and abstraction layers, the interconnection of interfaces between the provider domains can be archived in many ways [4]. From an orchestration layer perspective, the most common approach is to have a top-level single component that orchestrates subsystems. However, in addition to interconnecting the Service Providers through a common orchestration plane, it is also possible to offload the orchestration plane by interconnecting domains at other abstraction layers.

One method to interconnect Service provider domains is synchronizing the control planes between the providers to allow them to be perceived as one. For network control systems, all the network resources are perceived as one single SDN pool, while, for NFV, the Network Function Virtualization Infrastructure (NFVI) is perceived as one single pool [5] as well. Examples of such technologies are SDNi [6] or the Border Gateway Protocol (BGP). The problem with these methods is the dependency on the data plane, where different domains use different routing protocols or different types of SDN controllers. Another method is to provide an additional control plane abstraction layer that translates different control plane protocols to one standard such as the Forwarding and Control Element Separation protocol (ForCES) [7] or the Control Orchestration Protocol (COP) [4]. This introduces an additional overhead and also requires the involved partners to support the abstraction layer standard.

The European Telecommunications Standards Institute (ETSI) suggests interconnecting multiple domains on the Management and Orchestration layer with less focus on an east–west control plane protocol [8]. Furthermore, they aim to let the orchestration plane configure multiple control planes as they are multiple autonomous systems. Implicitly, ETSI aims to minimize the use of an east–west control plane protocol, while allocating the network intelligence and service routing on the orchestration layer. Their architectural guidelines [8] do not exclude a control plane to control plane protocol. However, a control plane protocol gives other opportunities that the orchestration layer does not support. Kulkarni et al. [9] shows that a control plane protocol such as Network Service Headers does provide an independent service plane, it opens up for exchange metadata of VNFs, and it enables the possibility to classify and tag packets independent of the other packet

headers. This also aligns with the need for SFC headers when running encryption as stated in the Introduction section (Section 9.1.1). SFC protocols that support additional SFC headers such as Network Service Headers (NSH) and Multiprotocol Label Switching (MPLS) [2] can be used in combination with an underlay of existing encryption protocols such as Internet Protocol SECurity protocol (IPSec). If the SFC header is preserved unencrypted along the SFC, it is possible for the routers to do SFC routing decisions, according to the control information contained in these headers. This makes it possible to run encryption services in front of the VNFs and still preserve routing information in the data packets.

The idea of using such a control plane protocol reflects the work from the Internet Engineering Task Force (IETF), where BGP is used for exchanging route information for NSH [10]. This NSH BGP control plane specification from the IETF lays the foundations for the architecture presented in this paper. However, BGP does not yet contain any information about the setup of encrypted SFC channels, while it offers no details about how the integrity of the attributes can be protected [11]. We extend this specification by introducing new encryption attributes to BGP and new Key Management Services (KMS).

IETF has suggested one expired Request For Comments (RFC) draft on a mechanism that supports the integrity of NSH headers [12], but this does not support per-flow encryption nor automation between multiple domains. Therefore, we aim to extend the NSH header integrity check approach by introducing additional upper and lower control plane channels for scaling, automation and encryption. From the encryption perspective, no new protocols have been found for encrypting SFCs. However, associated encryption technologies have research potential. An upcoming technology such as the Software Defined Internet Key Exchange (SD-IKE) [13] opens up new possibilities by running individual encryption per flow that is controlled by an SDN controller. Currently, SD-IKE is specified for the use in a single controller domain only, enabled by OpenFlow (OF). It does not work between two different SDN domains because of a missing control plane protocol and the lack of common SFC flow identities [14]. However, if an SFC-aware control plane protocol between an SDN controller is developed for this technology, it is possible to use one common encryption engine that encrypts every SFC flow individually. Due to the lack of inter-domain communication standards of SD-IKE, the architecture proposed in this article uses an alternative approach. We propose that each VNF is connected to a standalone encryption service that is only used once in one SFC.

No further research has been found that discusses per-user encryption of SFCs. Neither have any protocols or related research been found that supports encryption setup or key exchange mechanisms between multi-domain VNFs.

9.3 The Architectural Model

In this section, we present the proposed architecture at the highest abstraction level (Model), discussing the required entities, their relations, and their high-level functionalities. As presented earlier, the proposed architecture aims at SFC isolation enabled by automation of encryption channels. The architecture is based on nested SFCs, utilizing BGP to announce domain-specific information about network controllers and their respective encryption services. This information is again used to negotiate encryption services and keys for the purpose of securing the nested SFCs, maintaining a clear distinction between packet forwarding and tunnel configuration. Hence, the main components in the proposed architecture are:

1. Data-plane components for transitive SFC classification and forwarding. The SFC specification refers to these components as Classification Functions (CF) and Service Function Forwarders (SFF), that needs modification to support nested SFCs.
2. Control plane components for information sharing, with BGP and key distribution for encryption setup. The main components here consist of a Software Defined Network Controller with the BGP capabilities.
3. Management and Orchestration (MANO) applications, in order to orchestrate and provide encryption services to automate the set up of VNF isolation.

The architectural model in Figure 9.5 exemplifies how VNFs in an SFC can be isolated and encrypted in accordance with the proposed architecture, conforming to the SFC specifications [15]. This example reflects data plane packet forwarding of four Service Providers in a modular SFC. In summary:

1. The incoming data packets of the Voice Over Internet Protocol (VOIP) and the HyperText Transfer Protocol (HTTP) are classified according to the specific (VNF A>B>C>D) SFC path. Since the SFC path is predetermined by distributed route tables, the SFC headers are added to the packets.
2. Due to VNF isolation requirements, the packets are classified and forwarded based on two layers of SFCs. Hence, the first classifier is also adding the second isolating SFC header. For this example, two inner SFCs are established: one for VNF A>B>D–HTTP components and one for VNF A>C>D–VOIP components. This ensures that, with respect to routing, VNF B and VNF C are isolated from each other.
3. To ensure the encryption of the packets, each hop in the inner SFC paths must be encrypted. Hence, the network controller is distributing the SFC paths to traverse sets of pairwise encryption services. The network controller is also distributing the encrypting keys per link.
4. The packet forwarding continues with consecutive encryption > processing

> decryption sequences, in accordance with a distribution of pairwise keys among the providers, and with SFC header modification per hop. Pushing and popping of additional SFC headers ensure that the SFC path is maintained. The SFC path is therefore an end-to-end encrypted tunnel, implemented as interconnected chains of hierarchically encrypted links, where, in the presented example, providers A-B-D can access only the HTTP component, and providers A-C-D only the VOIP component of the SFC.

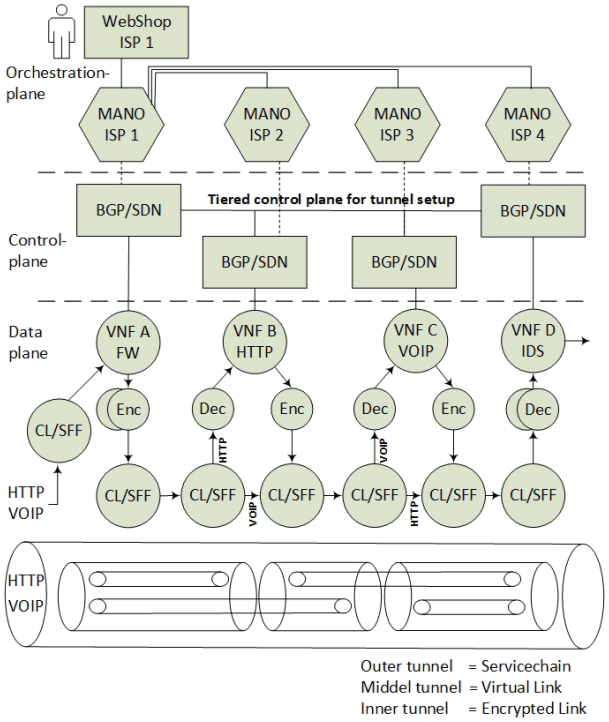


Figure 9.5: The architectural model.

In order to provide the required functionalities for these operations, the following modifications have to be integrated into the overall architecture.

9.3.1 The Data-Plane—A Hierarchy of SFC Headers

To be able to isolate and encrypt traffic between different VNFs, an encryption service has to run at all ends of every VNF in the SFC. Furthermore, to ensure compatibility with the SFC specification [15], the encryption services are separated from the VNF and the Service Function Path (SFP) itself. Hence, the proposed architecture introduces additional encryption tunnels in the SFC model, which are identified as inner SFC tunnels, and defined by the SFC header 1 (Figure 9.6),

while SFC header 2 corresponds to the original SFC header in accordance with the SFC specifications [15]. It must be noted that the Virtual Links (VLs) are layered headers and are not implemented as communication tunnels, but they can be perceived as virtual tunnels from an architectural perspective. In practice, the SFC layers are placed below the IP layers in order to enable IPsec encryption. As previously mentioned, this is because encryption of an IP packet with SFC headers inside would hide both SFC header and the classification data [16].

We define the SFC header 1 to constitute the Encrypted Link (EL) and to let the SFC header 2 define the Virtual Link (VL), while the transport layer is named the Transport Link (TL). All of these packet headers need routing information associated with them, which is defined by a tiered control plane (see next Section). From an OSI-model perspective, we define the two new SFC layers to belong between layer two and layer three (Figure 9.6). The Transport layer is according to the SFC specification [15] referred to as the layer that transports SFCs. Here, we define the transport layer to be an IPsec tunnel.

The SFC header 1 is always associated with an Encrypted Link that consists of an inner SFC with one hop only. Due to the static nature of the inner SFC header 1, this information is placed as an extension of the original SFC header 2 and not a next-protocol header. For SFCs enabled by NSH headers, this means that one NSH header can contain both SFC 1 and SFC 2 headers. Section 9.6 shows how the original SFC header of NSH is extended to include inner SFCs identifiers by introducing a new type value of the NSH header.

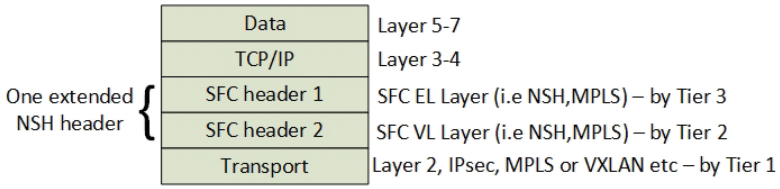


Figure 9.6: Additional Service Function Chaining layer.

9.3.2 The Control Plane—Tiered Tunnel Automation

To enable the automated creation of the encryption tunnels, a common control plane is required across the VNF Providers. This is similar to the current use of BGP as a global common control plane for internet traffic routing among Internet Service Providers. Such a control plane for NFV service chained traffic must support the exchange of service capabilities, flow-specific routes, and service chains. In principle, keys and VNF associations must be distributed between every Service Provider contributing to the SFC, in a per-hop > per-NFV service > per-user basis,

while sharing such information globally imposes scalability limitations.

Therefore, the control plane is defined architecturally as a three-tiered control channel with a key set-up mechanism. Figure 9.7 presents the tiered control channels within the control plane, between the providers A and B of the previous example. This tiered concept follows the IETF SFC [15] standardization and the BGP control plane RFC for NSH [10]. However, it extends the functionality by introducing three tiers of control plane routes that reflect the hierarchy of data plane headers. Next, Tier 1, 2 and 3 functionality is explained.

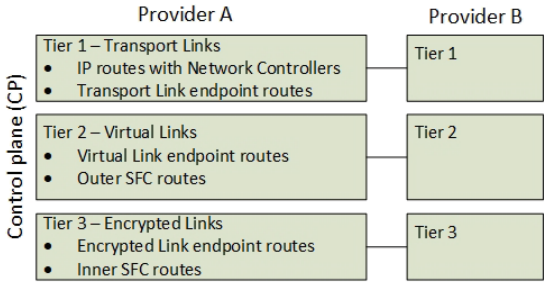


Figure 9.7: Multiple levels of communication channels.

Tier 1—Datacenter Sharing

The first tier exists in a global IP Virtual Private Network (IPVPN) where the NFV providers share their control plane attributes such as the network controller and their external Transport Links (Figure 9.7). The address of each network controller is announced to all peer controllers in the VPN, along with routing capabilities and connection properties in order to define how to connect to them. This information about the association between domains and their Transport Links is required in order to ensure routing support of SFC headers between the domains. In addition, since encrypted tunnels to a destination VNF are configured on the control plane, each controller must know which controllers to connect with, in order to set up the Virtual Links (VL) and consecutively inner Encrypted Links (EL).

For this purpose, every peer in the VPN serves as a proxy for the aforementioned information to others with normal BGP route distribution algorithms, so this information can be further used to set up a full mesh of Transport Links between datacenter domains (Figure 9.8). Figure 9.8 also shows an example of Provider A and C establishing a Transport Link between them based on the route information sent by proxy from Provider B.

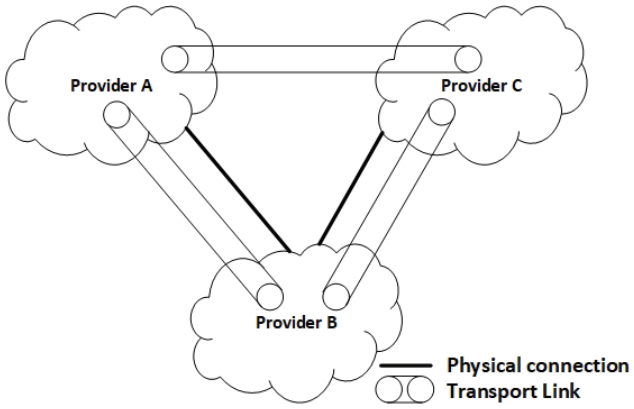


Figure 9.8: A full mesh of Transport Link tunnels.

A new multi-protocol BGP (mBGP) [17] address family is required because the BGP peers announce new types of IP routes that must not be mixed with regular BGP IP routes. This applies both to Transport Links and Network Controller routers. Consequently, new Address Family Identifiers (AFI) [17] and new Subsequent Address Family Identifiers (SAFI) [17] have to be defined by the Internet Assigned Numbers Authority (IANA), to ensure a global identification of which IP routes a network controller is responsible for. Furthermore, the use of a standard Resource Public Key Infrastructure (RPKI) [18] with BGP to secure the origin of the BGP speaker is required for distributing the public keys of each network controller, in order to provide identification, confidentiality and integrity of the network controllers and distributed routes.

Tier 2—The Announcements of VNF Locations and SFCs

The second Tier is a full mesh of one-to-one control channels between all network controllers involved in an SFC. The channels are utilized to exchange information about the route locations of the VNFs that reflect the SFC header 2 routes. The BGP announcements are split into two parts. The announcement of the location of the VNFs and the announcement of the SFC.

The full mesh of Transport Links allows BGP route distribution of flow-specific routes to be scaled down, since the routes are transmitted only to relevant network controllers, according to the network topology established in Tier 1.

Tier 3—The Announcements of Transitive SFC Routes and Encryption Service Locations

Tier 3 is responsible for establishing and managing the inner SFC Encryption Links. The route distribution is similar to Tier 2 and is also a process consisting of two parts. The announcements of the location of the encrypting services and the SFC announcements of the next hops in the SFC. These routes are announced over a separate Tier 3 BGP peering interface. The Tier 3 routes must be clearly distinguished from Tier 2 routes. Hence, a new BGP address family (AFI and SAFI) is also suggested to be defined for this Tier. The components of Tier 3 are therefore similar in nature to those of Tier 2, but they serve a distinct purpose and therefore must refer to a new address family.

The SFC header for Tier 3 is in this context not SFC subsystems such as nested SFCs where the inner SFCs belongs to a sub-chain of SFCs in RFC 7665 [15]). The Tier 3 SFCs are transitive, meaning that they always coexist with the upper SFC layer and contribute to routing decisions for both Tiers 2 and 3. To differentiate SFC subsystems from inner SFCs, the Tier 2 SFC header must also contain information about the type of the next SFC header. When using NSH, this means that both inner and outer SFCs must exist in one NSH header. Hence, it is a header extension and not a next protocol header.

Finally, an important prerequisite is that the encryption services are already running before the announcement of the Tiers 2 and 3 SFCs. Hence, Tier 3 SFC announcements need to be announced before Tier 2 SFC announcements. This also reflects the fact that information about Tier 3 SFCs must exist in Tier 2 SFCs, which is explained in the Protocol and Interface Section 9.5.

Encryption Automation

The control plane is also responsible for the setup of encrypted channels between the VNFs. In this architecture, we suggest automating the exchange of encryption keys. We define that it is the origin Service Provider in an SFC that is responsible for key distribution and tunnel automation. A Key Management Server on the control plane is defined to distribute the setup of the secure channels. The key distribution depends on the SFC and is therefore dependent on the order of the VNFs and how they are routed. Hence, a key distribution protocol that supports dynamic endpoint configuration and the negotiations of encryption keys from a trusted third party authentication server is needed. Section 9.5 suggests a simplified prototype of such a protocol where two random endpoints contact a third party server to receive instructions on how to establish a Security Association for IPsec.

9.3.3 The Management and Orchestration (MANO) Plane

To enable the automated VL encryption and VNF isolation with network redundancy, the architecture suggests having the network intelligence in the control plane, which implies that the Management and Orchestration plane (MANO) is less declarative about the network configuration. Instead of relying on the MANO to instruct the control plane about the locations of the VNFs and their corresponding encryption services, the control plane directly utilizes BGP and Key Management Services in order to configure the network dynamically. This enables the MANO to be less declarative about the network configuration, and enforces the control plane to have imperative services that can be dynamically reconfigured based on physical changes without MANO dependencies.

We simplify the ETSI reference model and only focus on new top-level services on the orchestration plane. The services include enabling VNF isolation and encryption that are the only parts that are relevant for the control plane components. Hence, the architecture is focused around the control plane where the ETSI MANO components such as the Virtual Infrastructure Manager (VIM), the VNF Manager and the orchestrator are not taken into account. The NFV MANO components are aggregated into one orchestration plane application, which enables a set of service requests, which are discussed in the next section.

9.4 Services in the Architecture

This section discusses the services that constitute the proposed architecture, beginning with the end-user services enabled by the orchestration plane, followed by the service components on the control plane and the data plane.

9.4.1 Service Components on the Orchestration Plane

The end-user orders a set of services (VNFs) from the ISP in a web portal, and while being unaware of the data-center location of each service, he wants to ensure that integrity and confidentiality are preserved in the SFC between the VNFs. The end-user requires both checking if the infrastructure is capable of delivering the services and also placing an order of service provisioning. This implies that the end-user can do four new types of service requests provided by the orchestration layer with respect to isolation and encryption: 1—Request of encryption capabilities per VL, 2—Request of isolation capabilities per SFC, 3—Request of provisioning an SFC with encrypted links and 4—Request of provisioning transitive encryption services to enable VNF bypass in the SFC (Figure 9.9).

These simplified end-user services are utilized in an application such as a webshop in the Operation Support System (OSS) domain (Figure 9.9). The practical

result of such service requests ends up in a list of VNFs connected to an SFC, which also includes an inner SFC specification. These are called Network Service Descriptors (NSD) [19] and are stored in a repository. This architecture focuses on hierarchical SFCs and encryption services, assuming minimal modifications of the ETSI model in the form of NSD extensions. However, modifications of the NSDs are required in order to support a description of encryption services (encrypting VNFs). Additionally, an extension of the NSD describing the VNF Forwarding Graph Descriptor (VNFFGD) is needed, in order to support hierarchical SFC descriptors. A type extension of the Virtual Link Descriptor (VLD) is also needed to describe inner Encryption Links for Tier 3. The orchestration application provisions the VNFs towards all relevant Service Providers, as well as provisioning backup VNFs for redundancy, while it distributes the relevant identifiers during the VNF instantiation. Furthermore, the control plane is responsible for selecting what VNF instances will participate in the SFC. The necessary NSD extensions are described in the Protocol and Interface section (Section 9.5). In summary, the orchestrator maintains the following functionality:

- Interpret end-user requests of VNFs and SFC in order to create the NSDs,
- Calculate where it is most efficient to run the VNFs,
- Consider any VNF constraints,
- Verify that it exists a Transport Link between every Service Provider that participates in the SFC,
- Ensures that the encrypting VNFs are co-located with normal VNF during provisioning,
- Provision VNFs at all Service Providers,
- Generate a pass-phrase (PSK) per VNF instance to enable authorization of the VNFs. When the VNF is provisioned, this key is submitted as a VNF application parameter.

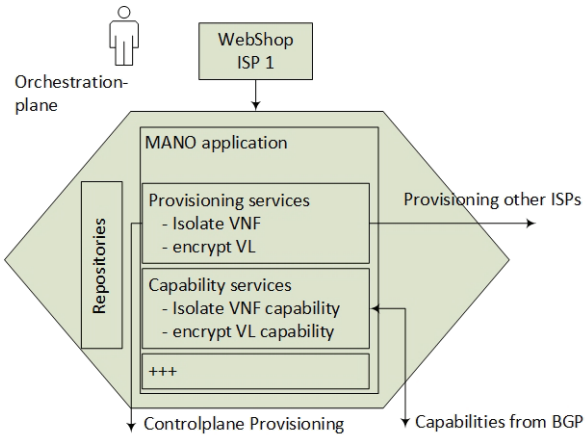


Figure 9.9: Services in the orchestration plane.

9.4.2 Service Components on the Control Plane

An interconnected multi-domain control plane protocol is the main contribution of this article. The most important services in the control plane protocol are the BGP services that exchange SFC route and the encrypted path information (Figure 9.10). This section explains the route distribution services and the surrounding control services.

The two main components of the control plane are the network controller and the Compute Node (Hypervisor) BGP services. The architecture is not based on traditional imperative SDN such as OpenFlow or distributed Open Virtualized Switches (OVSs) on the Compute Nodes, but it uses a declarative SDN method by the use of the BGP route distribution. Hence, the network controller is distributing SFC flow routes over BGP in order to inform all Compute Nodes about how to forward SFC packets. Every Compute Node announces its connected VNFs in order to let the network itself calculate the correct SFC paths. Similar to the BGP control plane for SFC [10], every Compute Node acts as a BGP speaker to announce its connected VNFs. Since the network controller knows the location of the VNFs, it can distribute the SFCs for both Virtual and Encrypted Links to every Compute Node. This section explains the functionality of this BGP service, how the edge VPN gateway connects the remote Service Providers and how the Key Management Server (KMS) can automate the setup of tunnels.

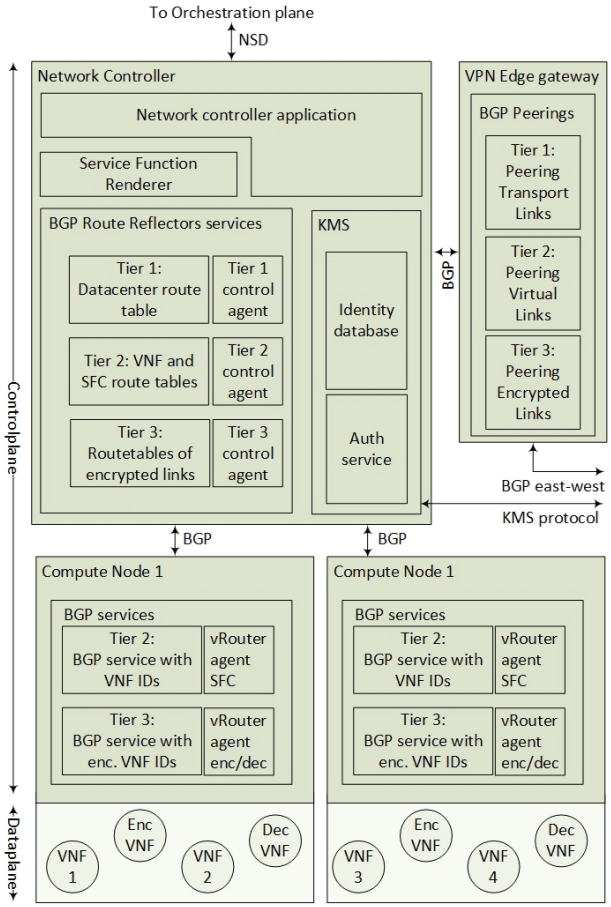


Figure 9.10: Services on the control plane.

The Network Controller

The network controller application is the main controller of all other services. It holds a control application and a Service Function Renderer that interprets the SFC provisioning. This architecture requires that the NFV orchestrators have provisioned all the VNFs, while the network controller is responsible for the network provisioning. According to the SFC RFC [15], the first SFC controller is responsible for setting up the SFC. The Service Function Renderer service selects the instantiation of an SFC path. Service Functions in an SFC may become unavailable or the physical constraints may alter the most efficient path. Hence, the SFC specified by the user can be different from the instantiated SFC path (aka Rendered SFC). For example, if a VNF is down and a policy allows the VNF to be bypassed,

then the corresponding Encrypted Links must also be bypassed. In this architecture, the network controller is simplified for supporting the following services:

- Mapping the instantiated VNF IDs to their Compute Node locations,
- Populating the KMS server with identities (see Section 9.4.2 - Distributed Key Management Services),
- Populating the Tiers 2 and 3 agents with the rendered SFC,
- Orchestrating Transport Links, Virtual Links and Encrypted Links,
- Recalculating SFCs for optimization or VNF bypassing during a network or Compute Node failure.

The Tier 1 BGP Service

A Service Provider establishes a peering with a remote data-center by a contractual Service Agreement. This agreement is established by an abstract level on the orchestration layer and is practically set up as BGP peering between the Service Providers. This article recommends having the BGP peering in a private IP-VPN, but, theoretically, it can be a public Internet connection with extended BGP features. Next, dynamic configuration attributes such as the network controller type, network controller address and data-center to data-center Transport Link are shared over this BGP peering. This information gets distributed to every BGP agent in the network. The second stage of the Tier 1 setup is to establish data-center to data-center Transport Links that are normally set up using VPN connections. A VPN link defines the Transport Link between data-centers, which ensures that the underlying network is transparent to packet forwarding and that does not require intermediate network elements to be able to read SFC headers. The architecture suggests using one VPN gateway to terminate all Transport Links.

From the SFC routing perspective, the VPN gateway is the next hop for SFC packets going to a remote data-center. The next hop is determined by the VNFs locations of which are distributed by BGP in Tier 2 (Section 9.4.2 - Tiers 2 and 3 BGP Peerings). For direct peerings, this VPN tunnel is optional. The main purpose of this BGP service is to inform all network controllers about remote Transport Links (Provider A must know that a Transport Link exists between providers B and C). This means that, if no VPN tunnel is needed, the Edge gateway still has to announce that the Transport Link is established in terms of a direct peering such as a direct cable. The Tier 1 agent can verify that the full mesh of Transport Links exists, while it holds a table of the domains (AS numbers) that it is directly or remotely connected. The Tier 1 control agent is responsible for:

- Sharing network information over BGP,
- Instantiating Transport Links,

- Populating the Tier 1 BGP with Transport Links,
- Serving an Application Programming Interface (API) towards the orchestration layer for end-user services regarding capabilities of encryption and isolation.

Tiers 2 and 3 BGP Peerings

The BGP control plane for SFC [10] intends to let every Compute Node announce their virtual services (VNFs) as Service Function Instantiated routes (SFIR). In order to let the network itself calculate the correct SFC paths, the Tier 2 control agent can inject the Service Function Path Routes as a Route Distinguisher (SFPR-RD) and distribute them to the Compute Nodes. This route distribution enables each virtual router agent in the Compute Node to calculate the next hop in the SFC. Hence, the Compute Nodes virtual Tier 2 agent can calculate and change the SFC headers Service Path Index (SPI) field, in order to reflect the next hop in the SFC.

By introducing an additional layer of Encryption Links (Tier 3), each Compute Node also needs to distribute information about their encryption services. The encryption VNF identities announced from the Compute Node we define as Service Function Instantiated Routes with encryption services (SFIR-E). Accordingly, the controller announces the corresponding SFCs that contain the routing information about the encrypted SFC and its relation with the outer SFC, namely the Service Function Path Routes with Encryption Route Distinguisher (SFPR-E-RD) (Figure: 9.11).

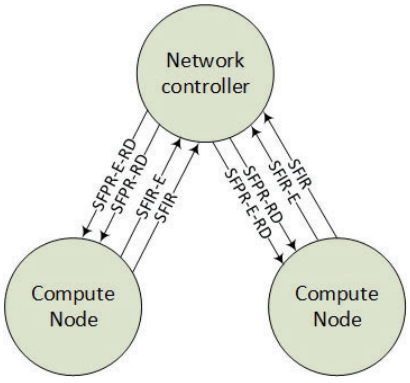


Figure 9.11: Border Gateway Protocol announcements.

The main roles of the Tiers 2 and 3 network controller agents, are to translate the rendered SFC into BGP routes. Since BGP also can contain redundant routes, the

redundant routes are also calculated and distributed according to what redundant VNFs have been provisioned. An additional functionality of the Tiers 2 and 3 control agents is to serve the encryption setup procedure with location information. The automated Encryption Link procedure is based on pre-provisioned VNF services that are configured to contact a common KMS service in order to exchange their encryption keys. Hence, the route messages of SFPR-E-RD must be populated with encryption information (public keys) by the Tier 3 control agents. The control agent makes sure that the VPN tunnel is set up before the route is announced by polling information from a Key Management Service. In summary, the Tier 3 control plane is responsible for:

- Distributing route information of how to route the SFCs to other Compute Nodes—this also includes redundant routes,
- Translating the SFC into BGP messages,
- Serving the KMS server with information from BGP, such as identities and encryption keys.

Distributed Key Management Services

One key problem in the current NFV architecture by ETSI [20] is that no trusted party is defined for the case when multiple Internet Service Providers want to agree on a shared pair of keys. Furthermore, there is no domain name system, in order to locate the VNFs in order to use URL identities and keys such as SSL. However, the BGP announcements enable the Service Providers to share a Key Management Server (KMS) and to announce the public keys for every peer. We define that the Service Provider that originates the SFC manage both the SFC and the KMS server for all VNFs in an SFC instance. We define the encryption endpoint identities as the endpoint routes, namely Tier 1—Transport routes from VPN gateways, Tier 2—SFIR and Tier 3—SFIR-E. The Key Management Server holds a mapping of the pair of routes that constitutes these links. In the example, the endpoint RD = AS3:82.147.30.1,211 is paired with the endpoint RD = AS4:82.147.30.2,212 and together they form an Encrypted Link. For Tier 3, the KMS server is responsible for pairing these endpoint identities and mapping them to a corresponding IPSec Security Association (SA). The control plane application populates an "identity table" with the basic information of the domain (AS number), the VNF ID and the VNF provisioned PreShared Key (PSK). Dynamic information such as the KMS server and the Compute Node locations are populated from BGP. When this information is in place, the KMS server populates the table with a certificate (Figure 9.12). Note*: Tier 2 encryption and VNF authentication are considered redundant if Tier 3 is in use. Hence, Tier 2 encryption can be skipped if Tier 3 is enabled.

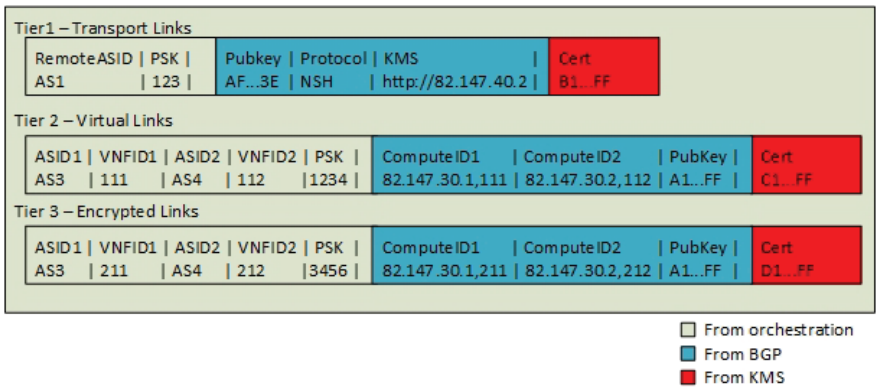


Figure 9.12: Key Manangement Service identities.

Tier 1 authentications use the remote AS as the authentication identity, while Tiers 2 and 3 use a concatenated string of domain (AS number), Compute Node and VNFID as the authentication identifier. For example, AS1:82.147.30.1,111. Note*: The global VNFID is not sufficient as an identifier since VNF migration to another Compute Node requires the keys to be changed.

For Tier 3 authentications, there is a clear KMS server authority that is assigned by the origin Service Provider in the SFC. Tier 1 authentications have, on the other hand, two potential KMS servers. By design, the first Service Provider that needs an SFC to a remote Service Provider initializes the connection. If both Service Providers instantiate a VPN tunnel simultaneously, two Tier 1 VPN connections can exist. BGP then automatically selects the most preferred transport route. Therefore, the Tier 1 control agent can optionally shut down the second tunnel.

The KMS controls all the keys in the architecture. It contains the primary keys that are set up during VNF provisioning and it derives dynamic keys for the setup of secure channels. Common for all tiers is that two pairs of public and private keys are used to set up a secure connection between the peer, while a shared secret (PSK) is used to authenticate the peers. For the Tier 3 encryption link, an additional SA is derived according to the KMS protocol explained in Section 9.5.2 - Key Management Service Interfaces.

Figure 9.13 shows a summary of the different keys that can be used. It shows that every VNF gets their unique certificate and authentication key during provisioning that corresponds to a unique KMS server certificate and an authentication key (PSK). The SA between the VNF and the KMS is established from these keys. Furthermore, the SA directly between the peers is instantiated dynamically over the existing SA between the KMS and the VNF. These are SA that are dynamically

derived from the primary keys from the KMS and VNFs.

The main functionality of the KMS server can be summarized as:

- Give two random endpoints (VNFs) instructions about how to set up a secure IPsec transport mode channel between them,
- Authorize both endpoints based on their identifiers, a PSK and their certificates,
- Serve the setup of Encrypted Links, Virtual Links and Transport Links by a key exchange protocol.

Furthermore, explanations of the KMS server protocol is explained in Sections 9.5 and 9.6.

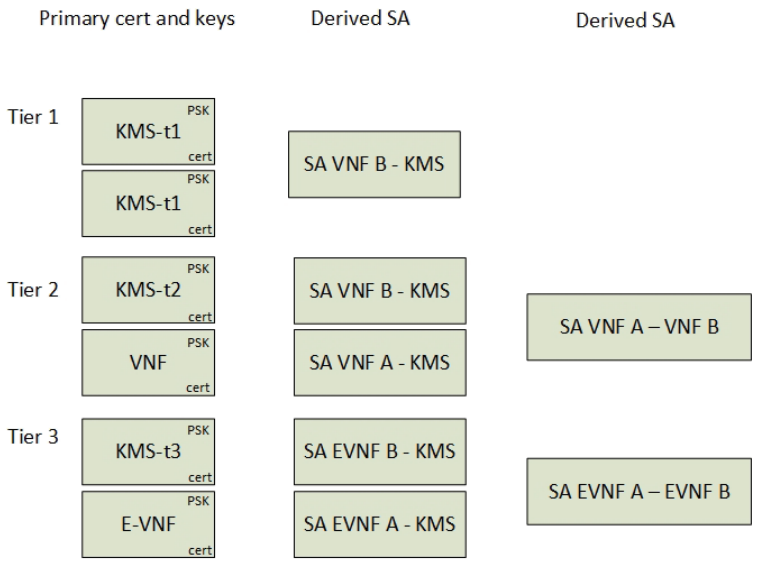


Figure 9.13: Overview of the encryption keys.

The Edge VPN Gateway

To enable the automated establishment of data-center to data-center Transport Links, an Edge VPN gateway is used for this purpose. Because this gateway has the responsibility of establishing Transport Links, it needs a southbound configuration interface towards the network controller. This configuration consists of a tunnel interface that includes a pointer to a KMS server. However, the VPN configuration can also consist of a full IPsec tunnel configuration defined by the orchestration layer or it can also be configured manually. The network controllers southbound configuration interface to the External VPN gateway is a domain

specific choice, where RESTconf [21] or Command Line Interface (CLI) are most common and recommended to use. Furthermore, if a VPN tunnel setup with a KMS server configuration is possible, this is announced over the Tier 1 BGP peering. The External VPN gateway contains BGP peering interfaces for all tiers. This enables both an exchange point of SFC routes between the Providers and it makes the VPN gateway capable of routing SFC packets.

It is optional whether the Tier 2 and 3 peerings are established over the VPN tunnel or if it is a multi-hop BGP peering. However, from a security perspective, the peering is more protected if it runs over the VPN connection. It is also optional if the Tiers 2 and 3 peerings run as one or two peering instances. Since they use different address families, it is preferable to use one peering. To enable the scaling of Tiers 2 and 3 routes, it is suggested to use a BGP route filter to only allow direct peers to be announced over the BGP link. This means that Provider A only receives Provider B SFIRs from the Provider B peering, where Provider B not will proxy Providers C SFIRs to Provider A. In summary, the Edge VPN gateway must be able to:

- Dynamically establish the Transport Link to other data-centers by terminating VPN interfaces,
- Route SFCs to the corresponding Transport Links,
- Filter SFIR and SFIR-E routes for relevant peers.

9.4.3 Service Components on the Data Plane

Figure 9.14 shows the data plane services running on the Compute Nodes. They consist of Classifier Functions (CF), Service Function Forwarders (SFFs), Encryption Services, VPN gateways and VNF applications, which will be further explained in this section.

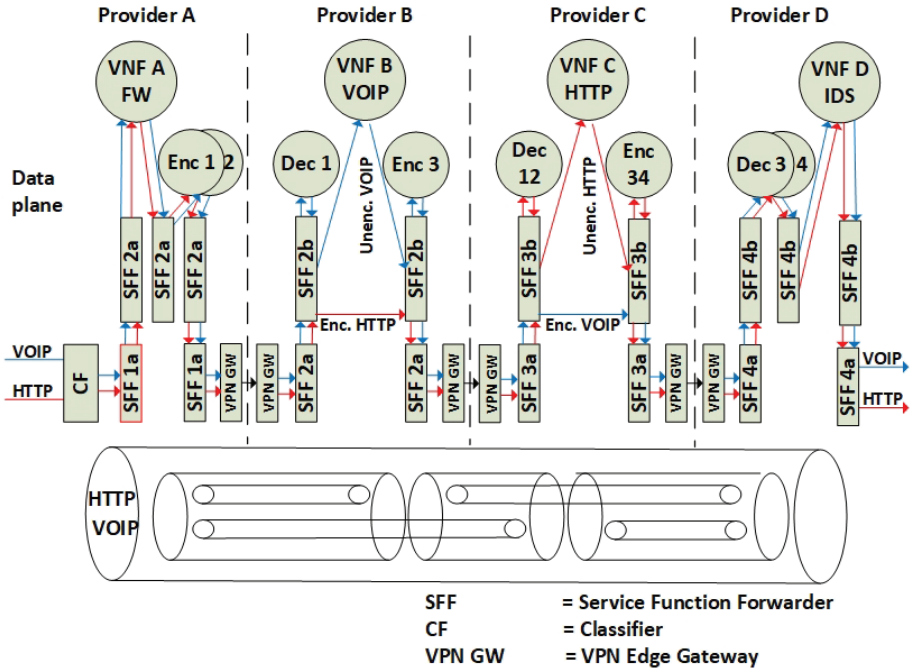


Figure 9.14: Services on the dataplane.

Classification

Before a packet enters the NFV domain, it must be classified according to what SFC it belongs to. Typically, the classification is based on source IP address and destination (TCP/UDP) port, but it can also be based on any attributes of layer 2 or layer 3 headers. The classifier adds the SFC headers to the packet, utilizing a look-up table of mappings between SFC identifiers and packet classification attributes. Because the packet encryption hides the TCP/UDP ports, this architecture assumes that it is only the classifier in the beginning of an SFC that does the packet classification based on non-SFC headers. Furthermore, SFC forwarding requires that the SFC header uniquely identifies the packet for each SFC hop. On the other hand, SFF proxies require that the SFC headers are removed before the data packets entering the service functions. In order to be compliant with SFF proxies, uniquely identifiable interfaces between the service functions and the SFFs contribute to the re-classification of SFC traffic that is traversing an NSH proxy.

The result of the classification is a combination of inner and outer SFC identifiers that corresponds to traditional classification rules based on IP. For example: Source IP 82.147.41.42 with destination port 80 is mapped to SFC header 1 with ID 10

and SFC header 2 with ID 20, while source IP 82.147.41.42 with destination port 5060 is mapped to SFC header 1 with ID 10 and SFC header 2 with ID 30.

Service Function Forwarder

The architecture reuses the principles from the BGP control plane protocol RFC draft [10] to let Service Function Forwarders (SFFs) be responsible for forwarding SFC packets and to announce their connected Service Functions. This RFC suggests using one SFF per Compute Node, but, since the architecture includes an additional SFC header, the SFF must also be able to read double-tagged SFC headers. Therefore, it is suggested to extend the RFC draft to include a hierarchy of SFFs per Compute Node. This is achieved by having one SFF to handle SFC header 1 forwarding decisions and one SFF to handle the forwarding of SFC header 2. Figure 9.14 refers to these two components as SFF1 and SFF2. Both SFFs send and receive SFC routes from a network controller over BGP and is responsible for forwarding packets containing SFC headers according to the revised SFC routes. This enables the SFFs to make dynamic forwarding decisions based on the SFC routes received from BGP. The hierarchy of SFFs on the same Compute Node implements the requirement [16] that SFC encryption services must be co-located with the VNFs. This makes it secure that encryption services for a VNF cannot be moved to a different Compute Node without also moving both the VNF and the encryption service together.

Encryption Services as VNF

The encryption service is implemented as a VNF in terms of a Virtual Machine or as a container application, but it can also be a separate service per Compute Node running as a hypervisor component. It is suggested that the application that runs inside the VNF is a simple IPsec service running in transport mode. The architecture assumes that not all VNF applications are able to read SFC headers. Hence, the encryption and decryption application expects incoming data packets to have the SFC headers stripped off. In addition to data plane forwarding, the VNFs must have the capability to be managed in respect of IPsec application configuration. ETSI suggests not using the Element Managers (EM) [19] for VNF configuration. Therefore, the architecture suggests having an out of band network interface to the VNF instances to handle the KMS protocol and the key management. This is implemented as a separate key management network interface in the VNF/Virtual machine. The IPsec application key management service is therefore not affected by SFC routing. The encryption application is preconfigured with a KMS identifier and PSKs as VNF startup parameters. Furthermore, Security Associations (SA) are derived from the KMS service (see Section 9.4.2 - Distributed Key Management Services). The type definition field in the NSD can be used to tag the

VNF as a special encryption VNF by following the standard NSD model (Section 9.4.1). However, a separate boolean field that defines whether the VNF is a traffic encryption service or not is recommended as a future NSD standardization.

9.5 Protocol and Interfaces

This section explains how the defined services communicate with each other and it specifies the most important interfaces and parameters that are exchanged between them. The orchestration plane, the control plane and the data plane interfaces are described by highlighting their main interfaces.

9.5.1 Orchestration Interfaces

The services on the orchestration plane are defined in the previous section (Figure 9.9). In this simplified architecture, we omit the webshop interface and parts of the orchestration plane provisioning. Hence, the interface to and from the orchestration layer is simplified to only include the functions needed for network provisioning. This includes provisioning SFCs, Transport Links and VNFs. Additionally, we have included the capability interface, which is required to obtain information about remote Transport Links between third-party Service Providers.

Provision SFC

The provisioning function interface receives an NSD file that we have formed as a simplified pseudo-YAML file format supporting both the ETSI NFV and the TOSCA [22] standard. Here, the messages are compressed in a simplified proprietary manner to visualize the content of the configuration exchange. Hence, the messages do not match syntactically with the YAML format (Figure 9.15). The relevant information elements in the messages are:

- The Virtual Network Function Descriptor (VNFD), which in this prototype describes the instantiated Virtual Machines global identifier (VNF-ID). The VNFD also includes a description of whether the VNF is a normal VNF or an encrypting VNF (EVNF). Additionally, it includes a new Preshared Key variable (Key), which is a field that must be standardized. This is suggested to be standardized in the TOSCA VNF Configurable Property name-space [22].
- The Virtual Network Function Forwarding Graph Descriptor (VNFFGD), which describes the SFC. The format of the VNFFGD needs to include both the inner and the outer SFCs. For Proof of concept purposes, we simplify the orchestration message to one new custom file descriptor as pseudo-YAML (Figure 9.15).

```
SFC-ID=12345:
 HOP: Type=VNF,VNFID=211,Domain=AS1, Key=9876,
        IngressVLink=InnerClassification,EgressVLink=VL1, E-SFC-ID={123451,123452]
 HOP: Type=VNF,VNFID=112,Domain=AS2,Key=9875,
        IngressVLink: VL1, EgressVLink: VL2,E-SFC-ID={123451,123452]
 HOP: Type=VNF,VNFID=113,Domain=AS3,Key=9874,
        IngressVLink: VL2, EgressVLink: VL3,E-SFC-ID={123451,123452]
 HOP: Type=VNF,VNFID=114,Domain=AS4,Key=9873,
        IngressVLink: VL4,EgressVLink: OUT,E-SFC-ID={123451,123452]

E-SFC-ID=123451:
    classificationrule=tcp80,
    INNERHOP: Type=EVNF,OUTERVNFID=211,Domain=AS1,VNFENCID=211, Key=2222,
                IngressEVLink=classification, EgressVLink=EVL11
    INNERHOP: Type=EVNF,OUTERVNFID=212,Domain=AS2,VNFENCID=212, Key=3333,
                IngressEVLink=EL11, EgressVLink=EL12
    INNERHOP: Type=EVNF,OUTERVNFID=212,Domain=AS2,VNFENCID=212, Key=4444,
                IngressEVLink=EL12, EgressVLink=EL13
    INNERHOP: Type=EVNF,OUTERVNFID=212,Domain=AS2,VNFENCID=212, Key=5555,
                IngressEVLink=EL13, EgressVLink=EVL14
    INNERHOP: Type=EVNF,OUTERVNFID=214,Domain=AS4,VNFENCID=214, Key=6666,
                IngressEVLink=EL14, EgressVLink=EVL15
    INNERHOP: Type=EVNF,OUTERVNFID=214,Domain=AS4,VNFENCID=OUTER, Key=7777,
                IngressEVLink=EL15, EgressVLink=EVL16
E-SFC-ID=123452:
    classificationrule=udp5060
    INNERHOP: ……..
```

Figure 9.15: SFC provisioning message from the orchestration layer.

Get Capabilities

This interface obtains information about the Transport Links. The response includes a list of the BGP tables for Tier 1 formatted also as NSDs.

Provision Transport Links

The orchestration plane receives a new SFC and calculates all Service Providers that participate in the SFC. If a Transport Link does not exist, the orchestration plane is responsible for setting up this link. If the Transport Link is set up manually, the orchestration layer informs the control plane in order to enable the BGP announcements of the Transport Link. If the link does not exist, an NSD is sent to the other orchestrator. This NSD can contain a full VPN configuration, but in the architecture the message is simplified to only contain the domain (AS number) and a Preshared Key (PSK). Furthermore, each control plane instructs the Tier 1 agent to set up the Transport Link by the use of a KMS server.

Provision VNFs

A VNF provisioning message goes between the providers' orchestration plane. This message is also constructed as NSD, but in this paper it is simplified to a pseudo-YAML format (Figure 9.16). When all Service Providers have provisioned their services according to the NSD, each Compute Node will inform the related controllers about VNFs' locations. Therefore, it is only the origin Service Provider that needs to know the SFC. The other controllers only provision the VNF. For the VNF encryption services, the Preshared Key (Key) is also attached.

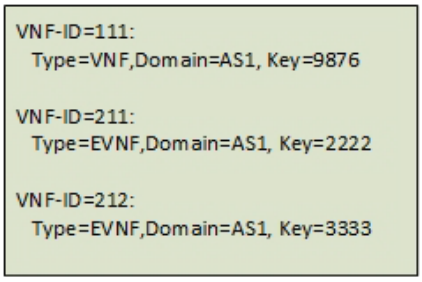


Figure 9.16: A Virtual Network Function provisioning message.

9.5.2 Control Plane Interfaces

The services on the control plane are defined in Figure 9.10. This section explains the interfaces between these control plane services.

Control Plane Application Interfaces

The control plane application is the main application in the architecture, but, from a service interface perspective, it only has two main interfaces towards the orchestration layer. Interfaces that the control plane application implements are perceived as interfaces held by other services, explained in the following sections. The main interfaces for the control plane application are:

- An NSD interface for incoming requests from the orchestration plane. This includes the NSDs for SFC, the VNFs and the Transport Links.
- A service capability interface to get information about the Transport Links to inform the orchestration layer whether the Transport Links exist and how they are established. This service is reflected from the orchestration plane and proxies the BGP route table to the orchestration plane as an NSD.

Tier 1 Interfaces

The Tier 1 BGP service sends BGP messages to other network controllers, while the control agent listens for service request for Transport Link maintenance (create, delete, get, modify). Additionally, the control plane application and KMS server reads the announced BGP messages, which means that the Tier 1 agents have three interfaces.

- A BGP speaker service running on the network controller. The BGP messages consist of two new address families. The new address families are reflected by the announcement of the network controllers and the announcement of the Transport Links (Figure 9.17). The address families are defined as Network Controller routes (NCR) and Transport Link routes (TR). These BGP messages are distributed globally.
- A configuration interface to inject new Tier 1 routes. The Tier 1 control agent receives a "create Transport Link" message from the controller application, and it injects a Transport Link route into BGP.
- A Get-Capability interface, which transports the BGP table to a YAML format that consists of all Transport Links.

Because of the tiered architecture of BGP announcements, the Tier 2 and the Tier 3 routes are automatically withdrawn if the Tier 1 Transport Link goes down. Hence, no further distribution of error handling messages is needed from the Tier 1 control agent.

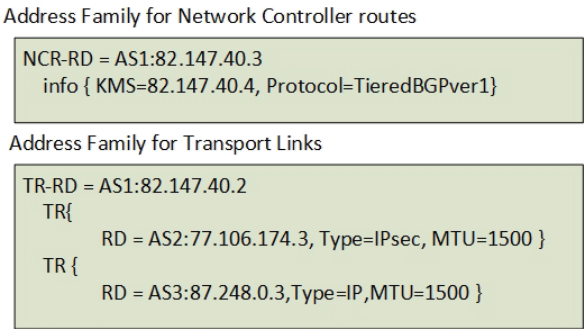


Figure 9.17: BGP announcements Tier 1.

Tiers 2 and 3 Interfaces

The Tier 2 and the Tier 3 control plane interfaces consist of BGP messages previously explained in Section 9.4.2. Figure 9.18 exemplifies how the original SFPR-RD messages [10] are changed into two new versions of SFRP-RD (Tier 2) and SFPR-E-RD (Tier 3) messages. The SFRP-RD message contains all the SFC hops

in the SFC and describes the inner Encryption Links as SFPR-E-RD inner hops. Each hop consists of a Service Index (SI) that is decremented for every hop. The Route Distinguisher (RD) globally identifies the Compute Node (AS number + IP address) and also contains a Service Function Identifier (SFI) that defines the VNF instance ID (i.e., AS1:82.147.36.200,3). This RD is also the global VNF identifier used for authentication (see Section 9.4.2 - Distributed Key Management Services).

The setup of the Tier 2 and the Tier 3 peering between network controllers and Compute Nodes are considered domain specific and assumed as manually provisioned. Tiers 2 and 3 control agents on the Compute Node contain a domain specific application interface that enables the attachment and detachment of a VNF to the network, in order to announce the presence of a VNF on the Compute Node. The control agents on the network controller correspondingly hold an interface that listens for incoming rendered SFCs. In summary, the interfaces to the Tier 2 and 3 services are:

- BGP speakers on Compute Nodes that announce connected VNFs (SFIR and SFIR-E).
- A BGP speaker on the network controller that announces the SFCs (SFPR-RD and SFPR-E-RD).
- Compute Node agent configuration interfaces for maintaining SFIRs and SFIR-Es.
- A Network controller agent configuration interface to maintain SFPR-RDs and SFPR-E-RDs.
- A Network controller agent interface that can transform YAML into BGP Tiers 2 and 3 routes and vice versa.

The VPN Gateway

The VPN gateway includes network protocol interfaces as follows:

- An IPVPN BGP peering interface peering towards one or more Service Provider neighbours.
- A VPN tunnel or a direct interface to all other Service Providers.
- A BGP peering interface towards the Tier 1 route reflector that announces the VPN links.
- A Tiers 2 and 3 BGP peering over the Transport Link.
- A configuration interface such as RESTconf or CLI to set up VPN links.
- A KMS server interface to accept VPN connections authorized by the KMS server.

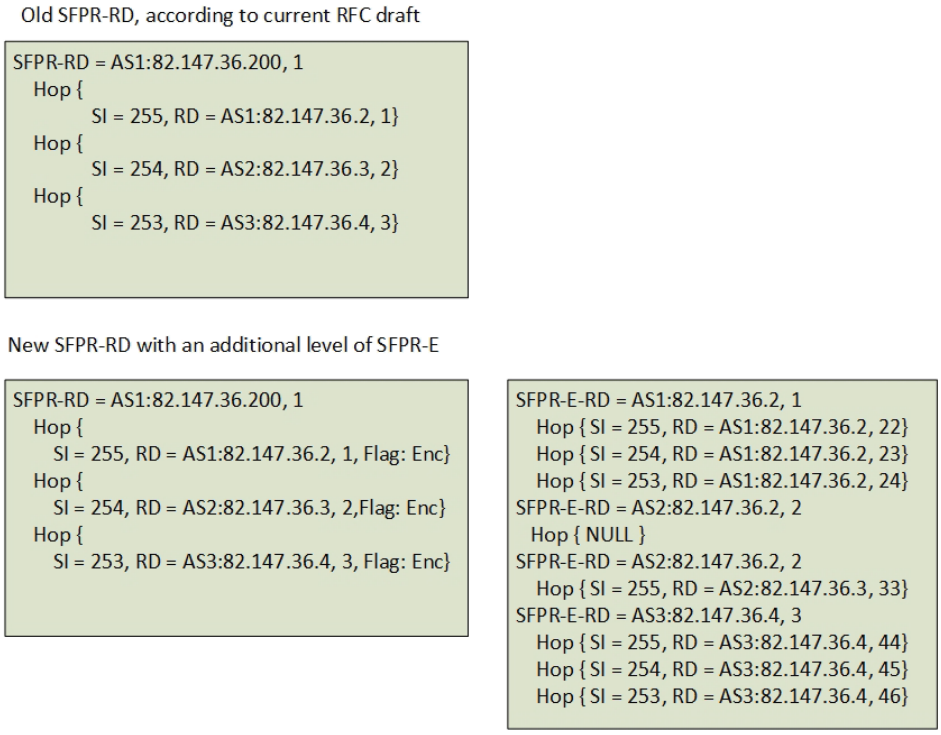


Figure 9.18: BGP announcement Tier 3.

Key Management Service Interfaces

The KMS server implements a protocol that can provide an IPsec Security Association (SA) between two VNFs running encrypting services. The protocol defines a trusted KMS server with two random endpoints (instantiated encrypting VNFs) as the client and the server, where the KMS instructs the VNFs to establish an SA. For the initializing phase, the KMS server and VNFs utilize a Public Key Infrastructure (PKI) to establish a connection between each other, which the Kerberized Internet Negotiation of Keys (KINK) protocol is referring to as PKINIT [23]. Hence, certificates are issued for peers by the use of public and private keys instead of using passwords. The public keys are distributed over BGP and secured by secure origin BGP (soBGP) [24]. The KMS service provides services such as ticket granting to ensure the integrity of messages to the server. To ensure a two-way authorization, an additional Preshared Key (PSK) authentication is added (Figure 9.19) to the protocol. The PSK is pre-provisioned by the orchestration layer. The KMS server protocol follows the same procedure for Tier 1–3 authentications, where the "user identity" is the only difference (see Section 9.4.2 - Distributed Key Management

Services).

An additional feature to the authentication and key negotiation protocol is the capability to inform the endpoint about the IP-address of the remote endpoint. After authentication of the endpoints, the remote endpoint address together with a new dynamic shared key is offered to the endpoints by the KMS server. Next, the endpoints establish a direct link between for the SA negotiations. Furthermore, implementation guidelines about the KMS protocol are given in Section 9.6.

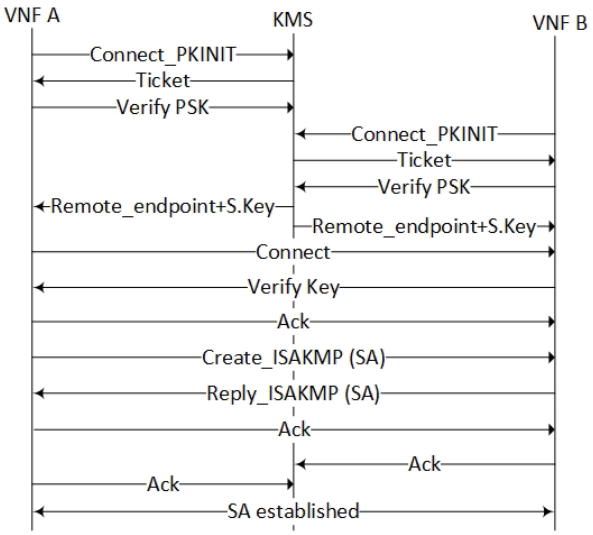


Figure 9.19: The KMS protocol (simplified).

In this architecture, the KMS server is simplified to only include one instance. In real life deployment, the number of KMS instances should reflect the number of control plane tiers. In summary, the KMS server holds two service interfaces:

- An authentication protocol interface used by the encryption services and the VPN gateway,
- A management interface to maintain the "user" identities (Section 9.4.2 - Distributed Key Management Services) and their corresponding PSKs.

9.5.3 Data Plane Interfaces

The services on the data plane have almost no relevant communication interfaces to other service components in the architecture. This reflects the clear separation of the control plane and data plane functions. The only component that communicates with the control plane is the VNF encryption service. This functionality

is explained in the KMS server interface Section 9.5.2 - Key Management Service Interfaces.

9.6 Implementation Guidelines

The tiered packet forwarding model of SFCs is considered as the main contribution in this article. The first phase of a proof of concept implementation is therefore only applied to the SFFs on the data plane, in order to verify the packet forwarding mechanism. Hence, a full implementation of the control plane, the orchestration plane and the encryption functionalities are omitted. However, relevant implementation guidelines are given for selected components of the architecture. This section also presents a procedural example that highlights the underlined functionalities.

9.6.1 Data Plane Implementation

Currently, no Virtual Infrastructure platforms support SFFs with double-tagged SFC headers. For Virtual Infrastructure systems with Virtual Extensible Local Area Networks (VXLANs) such as VMWare and OpenStack, the SFFs are implemented as distributed switches (i.e., OVS or VPP [25]) connected to distributed routers, where the VNF network interface is mapped to one VXLAN identifier. These platforms are currently neither capable of SFF forwarding nor announcing SFC headers over BGP with single or double-tagged SFCs. Therefore, it is suggested to use the RFC7665 [15] adoption principles by the use of an SFC aware SFF proxy to map SFC headers to interfaces such as VXLAN. This paper does not focus on adaptation services such as SFF proxies, but, for proof of concept purposes, an SFF proxy is needed to realize an SFF implementation on the Compute Nodes. The Fast Data–Input/Output (FD.io) framework [25] is used for implementing the SFF.

The open source FD.io framework provides fast and programmable IO services for networking and storage, while it can also provide the SFF functionality that is needed. A core component of FD.io is the Vector Packet Processing (VPP) library. This library enables implementation and testing of packet forwarding. An NSH-aware middlebox can be implemented in one (or multiple) VPP nodes, which represents an implementation of an SFF. SDN frameworks such as OpenDayLight (ODL) support VPP SFFs and opens up for testing the packet forwarding in further research of SDN and NFV control plane tests. This also makes it possible to utilize existing northbound interfaces such as the wrapper application named Honeycomb for ODL.

Figure 9.20 shows how an NSH header is formatted in order to support inner encrypted SFCs. We define a new type of NSH header named MDtype=3. This

header contains both the SPI of the outer SFC and the E-SPI for the inner SFC.

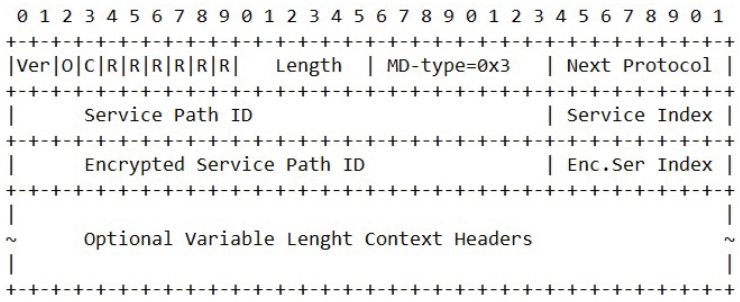


Figure 9.20: The Network Service Header structure.

Currently, the VPP FD.io framework does not support MDtype=3. However, the current version of the NSH standard includes other extension attributes that originally was intended to be used for passing information between VNFs. For proof of concept purposes, these extension attributes are used in a proof of concept implementation. Both NSH header types named MDtype=1 and MDtype=2 support such additional attributes. Figure 9.21 shows how an original NSH header with MDtype=2 and type, length, value (TLV) attribute extensions can be utilized to simulate the transport of E-SPI values.

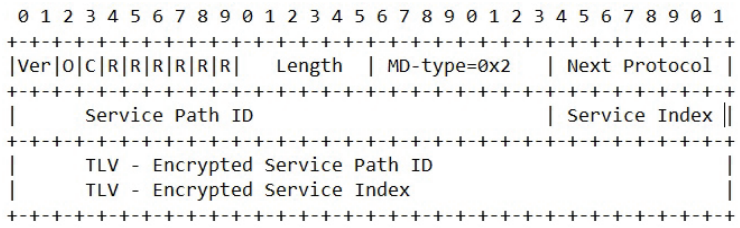


Figure 9.21: Packet structure in simulation.

Specifying the SFF forwarding rules for NSH are configured from the FD.io command line interface or through the APIs of the honeycomb application. For proof of concept purposes, we use statically defined FD.io command lines to configure the SFF to forward NSH packets.

Figure 9.22 shows examples of the CLI commands in FD.io VPP console application that configures forwarding of NSH packets. VXLAN tunnelling simplifies the forwarding between the SFFs and is used to interconnect SFFs within a domain. The NSH entry commands define the content of the NSH headers, while the NSH

map commands declare whether an NSH header is added (push), removed (pop) or modified (swap) during SFF forwarding. For MDtype=1, the NSH context headers are used to pass the inner SFC identifiers into NSH. MDtype=1 allows four context headers, where context header 1 (c1) contains the inner Service Path ID and context header 2 (c2) contains the inner Service Index. The nsp attribute in the vppctl command refers to the outer SFC identifier, while the nsi refers to the outer Service Index.

```
#Tunnel setup by VXLAN on SFF2
vppctl create vxlan tunnel src 192.168.6.121 dst 192.168.6.122 vni 1 encap-vrf-id 0 decap-next node nsh-proxy
vppctl set int l2 bridge vxlan_tunnel0 1 1
vppctl create vxlan tunnel src 192.168.6.123 dst 192.168.6.124 vni 2 encap-vrf-id 0 decap-next node nsh-proxy
vppctl set int l2 bridge vxlan_tunnel1 1 1

#VPP commands for setting up NSH entry and NSH push on SFF1
vppctl create nsh entry nsp 10 nsi 255 md-type 1 c1 100 c2 255 c3 0 c4 0 next-ethernet
vppctl create nsh entry nsp 10 nsi 254 md-type 1 c1 200 c2 255 c3 0 c4 0 next-ethernet
vppctl create nsh map nsp 10 nsi 255 mapped-nsp 10 mapped-nsi 255 nsh_action push encap-vxlan4-intf 3
vppctl create nsh map nsp 10 nsi 254 mapped-nsp 10 mapped-nsi 254 nsh_action push encap-vxlan4-intf 3

#VPP commands for setting up NSH swap on SFF2
vppctl create nsh map nsp 10 nsi 254 mapped-nsp 10 mapped-nsi 253 nsh_action swap encap-vxlan4-intf 3
```

Figure 9.22: Examples of the NSH forwarding commands for Vector Packet Processing.

It is emphasised that this utilization of the context headers does not support a clear differentiation from the normal NSH header and that the use of context headers for transporting inner SFCs only can be used for proof of concept purposes. Furthermore, analysis of a proof of concept implementation is provided in Section 9.7.

9.6.2 BGP Services

The architecture has suggested a wide range of new BGP address families and a set of new BGP services. The Internet Assigned Numbers Authority (IANA) has to assign new Address Family Identifiers (AFI) [17] and new Subsequent Address Family Identifiers (SAFI) [17] for the new protocol to be globally supported. For proof of concept purposes, it is suggested to extend an open source BGP service, such as Quagga [26], with these address family extensions and to make a wrapper application around the service in each domain that enables extraction and injection of BGP information. This wrapper application conforms with the virtual agents in the architecture. We suggested using Honeycomb for the SFF configuration and correspondingly Honeycomb can also control the BGP services on the SFFs.

9.6.3 KMS Server

The KMS server has similarities to the KINK [23] protocol that is based on Kerberos [27], but the KINK protocol does not have the support of additional PSKs and remote server connection instructions. For proof of concept purposes, it is suggested to extend the KINK protocol in the Racoon application [28] to also include a KINK-Validate-PSK() and a KINK-connect-Server() method (Figure 9.23).

```
Authenticate_KMS{
  KINK_Connect_KMS_PKINIT(ID,PubKey);
  KINK_Validate_PSK();
  KINK_Connect_Server();
  KINK_new_SA(TLID);
  KINK_establish_tunnel();
}
```

Figure 9.23: The Kerberized Internet Negotiation of Keys authentication protocol extension.

9.6.4 Control Plane Application

This section gives an example of how a tiered control plane architecture can automate the set-up of isolated and encrypted VNFs and how the control plane application can be implemented.

Figure 9.24 shows a subset of the steps in main procedure. It visualizes how a top-level contractual agreement between a set of NFV Service Providers can derive and set up subordinate control- and data-channels.

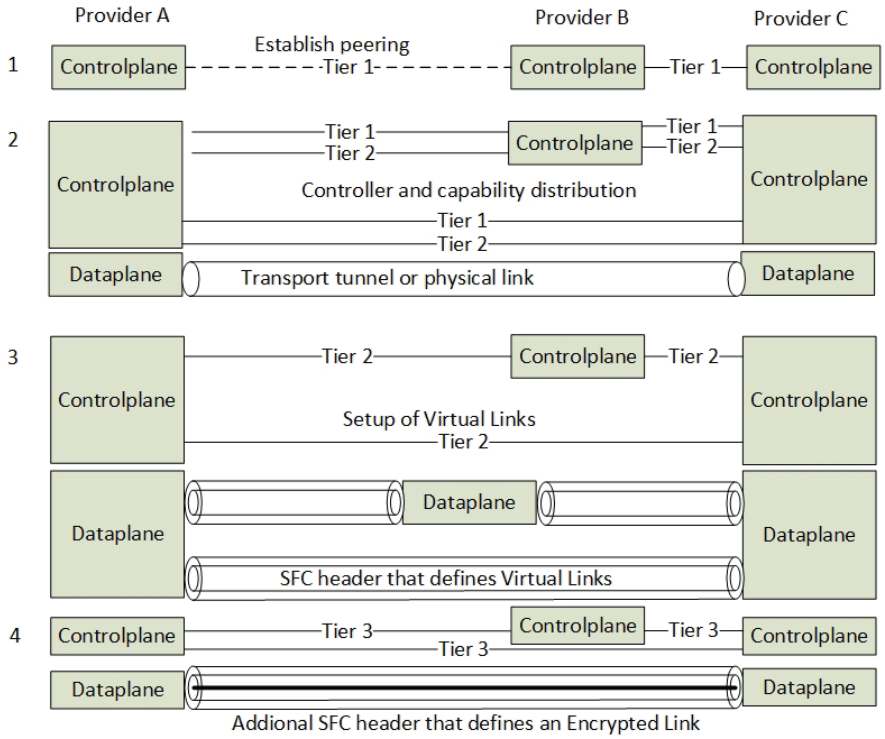


Figure 9.24: A visualization of the automated procedure.

Procedure example:

***Step A*—**A new link is established between Provider A and Provider B. A contractual peering is established on the orchestration layer and a BGP peering is made on the control plane (Figure 9.24(1)).

***Step B*—**The network controllers share their connection properties as Tier 1 attributes sent over BGP.

***Step C*—**An end-user orders a set of VNFs included in an SFC. The orchestrator instantiates the VNFs at the involved Service Providers by sending the NSDs for the VNFs and the Transport Link. If the "Get-Capability" service does not resolve a Transport Link, the Transport Link setup is processed first.

***Step D*—**The network controller application updates the repository with a new Transport Link and inserts a new row into the identity database for Transport Links for both network controllers. Additional information such as public keys are extracted from BGP and also posted to the identity database.

Step E—The network controller applications send configuration messages to their VPN gateways as RESTconf or CLI. The configuration includes both the VPN configuration and the Tier 2 BGP peering parameters.

Step F—When the VPN tunnel is up, the control plane application requests the Tier 1 control agent to inject a Tier 1 Transport Link route. At this point, all network controllers know about all the Transport Links, and a full mesh of BGP peerings is up (Figure 9.24(2)).

Step G—The network orchestrator verifies that all the Transport Links are up and that all controllers run the same protocol. It sends the NSDs about the VNFs to every orchestrator that consecutively instantiates the VNFs and sends the VNF IDs to the network control plane. It must be noted that the SFC NSD is sent to the origin network controller only.

Step H—The network controller parses the SFC and stores the instantiated path in the Rendered SFC repository. For every link that needs encryption services, it stores an encryption identifier in the KMS identity database. This includes the Tier 3 SFIR identities only (according to Figure 9.12).

Step I—The network controller now waits for the VNFs to be provisioned by periodically read the BGP Tiers 2 and 3 route tables.

Step J—The Compute Node updates its BGP agent with the VNF ID that further injects SFIR and SFIR-E routes into BGP during VNF provisioning.

Step K—For every VNF that becomes ready, the network controller reads the SFIR and SFIR-E routes and updates the KMS with additional identity information. It also updates the Rendered SFC repository about the physical location of the VNFs.

Step L—The encryption VNFs connect to the KMS server and establish the SAs according to the KMS protocol.

Step M— When the origin network controller has noticed that all the VNFs are announced and that the KMS server has registered all encrypted links, it calculates the SFC. The SFC is sent to the Tier 1 and the Tier 2 control agents in the network controller. They convert the SFC into SFPR-RD and SRPR-E-RD BGP messages and inject them into BGP. These messages are distributed to every Compute Node and instruct them on how to route the SFC packets. These messages contain the SFC header identities that constitute the Virtual Links and the Encryption Links (Figure 9.24(3,4))

Step N— For an incoming packet to the Compute Node, the SFFs can now look it up in the BGP route table and calculate the next hop for both inner and outer SFCs

headers.

9.7 Evaluation and Discussion

This section presents a proof of concept demonstration of packet forwarding with NSH headers. We also present an architectural analysis related to the proof of concept demonstration, the scalability and the limitations of the architecture.

9.7.1 Proof of Concept Demonstration of Data Plane Forwarding

The proposed architecture emphasises the need for encryption automation and suggests a tunnel hierarchy-model in order to overcome the SFC security problem. A full-scale implementation requires a modification of a set of NFV components, BGP network protocol extensions and it requires a development of a new protocol for key exchange between VNFs. However, a simulation of the data plane forwarding is tested in order to show the feasibility of the architecture. We state that a proof of concept demonstration on the data plane is the most important evidence that is needed before further implementations of control plane components are executed.

The test was performed on a simple VMWare ESXi 5.1 host (In a testbed provided by Eidsiva broadband, Olso, Norway based on Hewlett Packard DL380G7). Seven Virtual Machines (VMs) were created to simulate NSH packet forwarding. Four of them were running as SFFs with the VPP FD.io virtual switch software and three VMs were set up as simple end-nodes sending and receiving ICMP packets. All VMs ran Ubuntu 16.04 with the network interfaces connected to one single virtual switch. The VPP FD.io software version v18.04-rc2 was installed on every VM acting as an SFF with the NSH plugin enabled. All of the VMs were set up with /30 interface addresses with additional VXLAN tunnels defining the links between the SFFs.

Figure 9.25 shows the lab topology. The lab is simplified in order to show that NSH headers with extended information about inner SFCs can be forwarded similar to normal SFC headers. The difference from a regular SFF configurations is that this new SFF configuration can split one outer SFC identifier into two SFC paths based on the inner SFC identifier. Figure 9.25 shows that SFF A is classifying incoming traffic from hosts 1 and 2 into one common outer SFC and two different inner SFCs. SFF A is further splitting the SFC traffic into SFF B and SFF C. SFF B and SFF C are acting as NSH proxies that simulate the role of encrypting VNFs. The NSH proxy does in this setup swap the inner SFC IDs, while it maintains the outer SFC IDs.

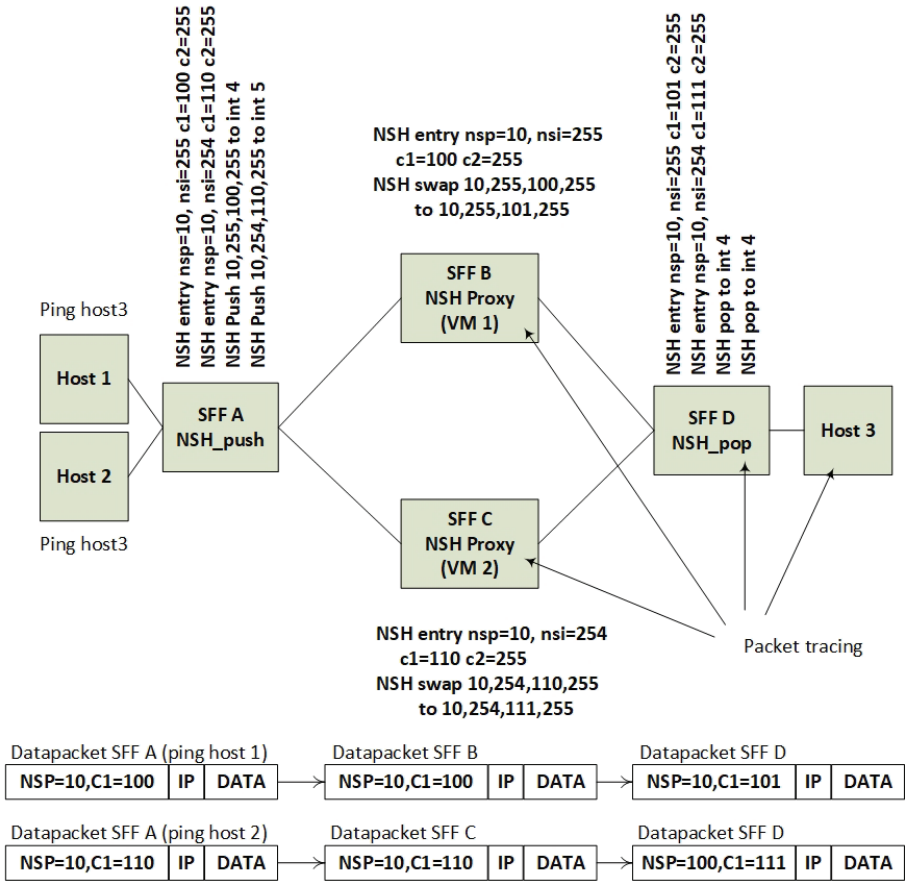


Figure 9.25: The lab topology.

Traffic observation by using the FD.io VPP packet capture and debug feature on every SFF verified that the NSH headers were classified and modified according to the topology (Figure 9.26).

It must be emphasised that this proof of concept demonstration is implemented with the current NSH standard. We have utilized the existing context header attributes of NSH to be used in a new context that can conflict with other use cases of NSH context headers. We have also utilized the outer NSH SPI and NSH SI attributes to define the classification of incoming NSH packets for both inner and outer SFCs. This is because classification based on context headers is not supported in VPP v18.04-rc2. Despite these adaptations, the demonstration shows that forwarding of NSH packets with a pair of SFC identifiers is feasible. However, one demonstration with statically defined configurations of NSH packet forward-

ing does not prove compatibility for all SFC topologies. It also does not verify how other SFC technologies such as MPLS forwarding comply with the architecture.

The required SFF forwarding functionality presented in Section 9.4.3 showed that the SFFs must be able to do forwarding based two sets of SFCs and additionally maintain the SFC header information along the packet path. A subset of this topology was implemented to show this core functionality of the SFFs. We simulated that SFF B and SFF C were connected E-VNFs, while SFF A and D only forward SFCs. The core functionality is performed by SFF A that is splitting the SFF traffic into two inner paths for encryption. The packet capture verified that this splitting of one outer SFC is possible while maintaining the outer SFC identifier.

The demonstration also shows that the SFC headers can be maintained along the SFC paths. It is assumed that if the inner and the outer SFC header were two different network protocol layers, the outer SFC header would have been lost during packet processing. However, our implementation of the NSH header contains both an inner and an outer SFC in one NSH network layer. This means that outer and the inner SFC headers do not need to be separated when the packet processing parses the different network layers. The proof of concept demonstration verifies that the information in the SFC headers is maintained during packet processing.

Discussion of Architectural Challenges

This section discusses a subset of the most important challenges that relate to the proposed architecture. The selected topics relate to the control plane and the service plane implementation challenges and discuss the constraints in the architecture.

```
00:17:43:493505: dpdk-input
  GigabitEthernet1b/0/0 rx queue 0
  buffer 0x185f4: current data 0, length 98, free-list 0, clone-count 0, totlen-nifb 0, trace 0x22
                  ext-hdr-valid
                  l4-cksum-computed l4-cksum-correct l2-hdr-offset 0
  PKT MBUF: port 3, nb_segs 1, pkt_len 98
    buf_len 2176, data_len 98, ol_flags 0x0, data_off 128, phys_addr 0x29817d80
    packet_type 0x10 l2_len 0 l3_len 0 outer_l2_len 0 outer_l3_len 0
    Packet Types
      RTE_PTYPE_L3_IPV4 (0x0010) IPv4 packet without extension headers
  IP4: 00:0c:29:4c:c1:87 -> 00:09:0f:a5:80:5d
  ICMP: 2.2.2.2 -> 2.2.2.3
    tos 0x00, ttl 64, length 84, checksum 0xed27
    fragment id 0x4579, flags DONT_FRAGMENT
  ICMP echo_request checksum 0x1841
00:17:43:493575: ethernet-input
  IP4: 00:0c:29:4c:c1:87 -> 00:09:0f:a5:80:5d
00:17:43:493581: l2-input
  l2-input: sw_if_index 4 dst 00:09:0f:a5:80:5d src 00:0c:29:4c:c1:87
00:17:43:493583: l2-learn
  l2-learn: sw_if_index 4 dst 00:09:0f:a5:80:5d src 00:0c:29:4c:c1:87 bd_index 1
00:17:43:493585: l2-fwd
  l2-fwd:   sw_if_index 4 dst 00:09:0f:a5:80:5d src 00:0c:29:4c:c1:87 bd_index 1
00:17:43:493587: l2-flood
  l2-flood: sw_if_index 4 dst 00:09:0f:a5:80:5d src 00:0c:29:4c:c1:87 bd_index 1
00:17:43:493588: l2-output
  l2-output: sw_if_index 5 dst 00:09:0f:a5:80:5d src 00:0c:29:4c:c1:87 data 08 00 45 00 00 54 45 79 40 00 40 01
  l2-classify: sw_if_index 1, table 0, offset c0, next 17
00:17:43:493589: nsh-classifier
  nsh ver 0 ttl 3 len 6 (24 bytes) md_type 1 next_protocol 1
  service path 10 service index 255
  c1 100 c2 255 c3 0 c4 0
00:17:43:493590: vxlan-gpe-encap
  VXLAN-GPE-ENCAP: tunnel 0
```

Figure 9.26: NSH packet capture.

Computational Overhead

The dynamic tunnel set-up and the data traffic encryption will increase the need for computational power. The massive amount of encrypted channels can both slow down the link performance and overloading data-center resources. To offload the service plane with encryption processes, a possible solution is to utilise a common encryption component to lower the resource consumption in a data-center. It is expected that a programmable data plane [29] and distributed containers [30] will be more available on enterprise switches and routers. Both of these alternatives can possibly make encryption services run more efficiently.

An MTU Increase

Encrypting IP packets often results in exceeding the Maximum Transfer Unit (MTU) and consequently ends up with packet segmentation. This is considered normal in many encryption set-ups. However, an introduction of another SFC layer can potentially make additional computational overhead and packet segmentation, which results in lower performance in packet forwarding. To prevent packet segmentation, it is possible to increase the MTU. However, adding the original NSH header can alone potentially trigger packet segmentation. If the NSH header length is constant, additional NSH attributes do not influence further packet segmentation.

This architecture suggests putting the SFC header between the OSI layer 2 and the layer 3. This means that it is the layer 2 or the SFC transport layer that need an MTU increase. Hence, it is the underlying Tier 1 transport that potentially can segment packets. General solutions to this problem are to introduce an MTU limitation in the setup of VNF encryption services or using Ethernet jumbo frames on the transport links.

Backup Tunnels and Resilience

The architecture opens up the use of backup tunnels in order to enable fast reroutes and multi-path SFCs. Making backup paths of all possible combinations of SFCs when a VNF fails creates an exponential set of extra tunnels and additional computational overhead. However, a set of bypass backup tunnels are assumed to not extensively impound computational power when they are not in use.

The size of the route tables depends on the SFC protection algorithm that is used. A simple link protection algorithm where only one VNF is allowed to fail linearly increases the SFC route table entries by the number of VNFs. A full mesh of redundancy link where many VNFs are allowed to fail increases SFC routes exponentially by the number of VNFs. Hence, link protection of SFC routes and SFC multi-pathing is suggested to be handled by the control plane. This is because a lower number of routes makes the control plane capable of scaling and to have low failover times. A full SFC protection is suggested to be handled by the orchestration plane by using re-instantiation of VNFs and redistribution of routes. This is a slow procedure, but it makes the solution scale better when all the possibilities of failures do not need any pre-calculation.

Encryption Key and Backup Keys Overhead

In comparison to a regular end-to-end encryption channel that consists of a shared key or a pair of keys, this architecture suggests using multiple encryption hops with multiple pairs of keys. The keys are primarily associated with the Virtual Link, where the KMS server holds the table of keys that is ready for use. Hence, the number of keys in use is directly connected to the number of Virtual Links, where each E-VNF that are defining the Virtual Links are associated with one key each.

Rearranging the order of VNFs in the SFC does not require any alterations of the distributed keys or any E-VNFs re-instantiation. The KMS server and the E-VNF maintain their Security Association even after an E-VNF to E-VNF tunnel is torn down. However, the Security Association between the VNFs must be re-negotiated when the order of the VNFs is changed. Since the KMS server controls multiple E-VNFs, it dynamically instructs the E-VNFs about where to connect.

The main key association is between the KMS server and the E-VNFs. The SA between the E-VNFs is dynamically created by the KMS server. Hence, the KMS server creates SA records in its database that correspond to the number of link protection channels (Section 9.7.1 - Backup Tunnels and Resilience). These database records create additional overhead, but it is not considered as a scalability problem. For example, an SFC that consists of four VNFs creates three additional link protection channels that result in three additional database records of SAs. Such additional database records are considered non-significant with respect to performance.

In addition to rearranging the order of the SFC, backup VNFs can also exist as redundant instantiations of the VNFs. From a control plane perspective, the behaviour of the KMS server is not changed, but it does consume more computational power. Typically, a pool of instantiated VNFs is instantiated together with a pool of E-VNFs. Both of these types of VNFs are registered at the KMS server where the control application makes a decision about how to connect the types of VNFs together. Hence, this overhead of additional certificates and authentication keys will have impact on the system, but it is not known how critical is for the the overall performance. Applying encryption in general clearly increase the need for computational resources (Section 9.7.1 - Computational Overhead). However, per SFC encryption automation has no scientific alternative and it is difficult to compare the performance to other solutions. In addition, a measurement requires a full-scale implementation of the system that our simulation study does not provide.

The Tier 1 encryption key is a one-time instantiation between the Service Providers are is not considered as key overhead. The Tier 2 encryption keys are only suggested to be used if no isolation is needed, and is expected to give an equal overhead impact to Tier 3 E-VPN.

The Dynamic Behaviour of VNFs

The architecture restricts the VNFs from altering the SFC headers in order to let the VNFs themselves decide the next hop in the SFC. In order to enable such functionality together with Encrypted Links, the E-VNFs must be pre-provisioned for every alternative route. Additionally, the VNFs must have access to information about what header attributes are available for modification, such as the next SFC hop. This enforcement of encryption is considered as a security feature since it restricts the VNFs from sending traffic to random VNFs. Corrupt VNFs can possibly both inject malicious traffic into other VNFs or make Denial of Service attacks towards other users. However, when a set of predetermined SFC paths is set, it gives the VNF a choice of multiple next-hops that network administrators and the network controllers have considered as secure paths. This also allows for

multi-tenancy VNFs, where a predetermined set of multiple users can share one single VNF. This functionality is not focused on in our work, but the architecture is considered to be easily adaptable for such scenarios.

Legacy Infrastructure

This architecture suggests a wide set of new protocol extensions to enable SFC isolation and automation. This requires that all involved Service Providers adapt to this protocol. However, the automation protocols are extensions of existing protocols, which allows the potential for old and new protocols to coexist. For example, the next header field in the NSH Base header must be set to a new type of NSH header in order to allow E-VNFs. In the proof of concept demonstration of the packet forwarding, this is set to 0×03. This capability of the SFF forwarding is announced by BGP Tier 1, but this Tier 1 capability information can also be exchanged manually. Hence, it is assumed that this architecture can coexist with legacy NSH infrastructures as long as the capabilities are announced over BGP.

With respect to non NSH networks, the Tier 1 transport tunnels ensure that underlying infrastructure between NFV service providers is transparent. An overlay network ensures that the network equipment not supporting NSH is bypassed by tunnels.

From a hypervisor perspective, the NFV infrastructure must support hypervisors that support the forwarding of NSH based E-VNF traffic. We believe that this research can contribute to a standardization of both a new NSH header extension and the corresponding BGP address families.

9.8 Future Work

For future work, we will proceed by testing the capabilities of the architecture. We have selected three areas of focus in order to make a full-scale verification of the architecture: the BGP control plane, the encryption services and the KMS key distribution protocol.

A tiered control plane with BGP can potentially raise the BGP convergence time, but since the BGP processes for Tiers 2 and 3 run in parallel, it is not known if the additional layer of SFC will significantly decrease the computational power on the control plane. In addition, it is not known how a full mesh of Transport Links will scale when the number of SFC routes and the number of interconnected NFV providers increase. Answering these questions will require a more extended test-case based validation of the proposed solutions, which is among our immediate plans.

Furthermore, the architecture suggests using VNFs on the data plane to enable

encryption services. To reduce computational consumption, it is possible to run one shared encryption application instead of multiple encryption VNF applications such as flow-based IPsec. To enable flow-based IPsec to work, the IPsec application must be capable of encrypting an IP packet without stripping or encrypting the SFC header. On the contrary, this SFC header introduces additional overhead that should be compared with the overhead of multiple encryption applications versus one application.

Another aspect on which we intend to focus for our future work relates to the KMS server, which is suggested to be implemented as an extended version of the KINK protocol. The KINK protocol is limited to only support IKEv1 and has limited support of future encryption algorithms. It is suggested to investigate protocol alternatives for key exchange between two random endpoints (VNFs) with a trusted third party KMS server.

9.9 Conclusions

In this paper, we presented an NFV architecture for SFC isolation and encryption in inter-domain NFV topologies. The architecture suggests using a tiered control plane protocol to distribute the SFC routing information as a control plane protocol standard between NFV domains. This is accomplished by using three tiers of BGP that both scale the route distribution and enable network redundancy. The tiers are reflected as Transport Links and two layers of SFC headers. The layered network model enables the possibility of encrypting and isolating the traffic in an SFC and ensures that encryption services must be co-located with the VNF. Furthermore, the use of a KMS server is suggested, in order to automate the setup of the tunnels between the encryption services. The architecture has been presented in four abstraction layers for completeness, namely: (i) Model; (ii) Service; (iii) Protocol and Interface; and (iv) Implementation. The main contribution of this article is the introduction of a new NSH extension header that enables such an architecture. A simplified proof of concept demonstration verified that extended NSH headers can be classified and forwarded in order to support an architecture that can isolate and encrypt SFCs. Our immediate future plans are focusing on the implementation details for the control plane components in the architecture. This includes a full-scale test-case based validation and verification of the proposed architecture.

Contributions

The main author of this paper is H.G.; V.G. and T.K. have contributed with respect to the paper structure, quality assurance and editing.

Funding

This research was funded by Eidsiva, the Norwegian Research Council and the Norwegian University of Science and Technology (NTNU).

Declaration of interest

The authors declare no conflict of interest.

References

[1] Paul Quinn and Uri Elzur. *Network Service Header*. Internet-Draft draft-ietf-sfc-nsh-13. Work in Progress. Internet Engineering Task Force, June 2017. 37 pp.

[2] Adrian Farrel, Stewart Bryant and John Drake. *An MPLS-Based Forwarding Plane for Service Function Chaining*. Internet-Draft draft-farrel-mpls-sfc-05. Work in Progress. Internet Engineering Task Force, Mar. 2018. 24 pp.

[3] Håkon Gunleifsen, Thomas Kemmerich and Slobodan Petrovic. 'An End-to-End Security Model of Inter-Domain Communication in Network Function Virtualization'. In: *Norsk Informasjonssikkerhetskonferanse (NISK): Bergen, Norway* (2016), pp. 7–18.

[4] Raul Munoz et al. 'Integrated SDN/NFV management and orchestration architecture for dynamic deployment of virtual SDN control instances for virtual tenant networks'. In: *Journal of Optical Communications and Networking* 7.11 (2015), B62–B70.

[5] NM Mosharaf Kabir Chowdhury, Muntasir Raihan Rahman and Raouf Boutaba. 'Virtual network embedding with coordinated node and link mapping'. In: *Conference on Computer Communications (Infocom)*. IEEE. 2009, pp. 783–791.

[6] Hongtao Yin et al. *SDNi: A Message Exchange Protocol for Software Defined Networks (SDNS) across Multiple Domains*. Internet-Draft draft-yin-sdn-sdni-00. Work in Progress. Internet Engineering Task Force, June 2012. 14 pp.

[7] Evangelos Haleplidis et al. 'ForCES applicability to SDN-enhanced NFV'. In: *Third European Workshop on Software Defined Networks*. IEEE. 2014, pp. 43–48.

[8] ETSI. *Network Function Virtualization (NFV) NFV-IFA 001 v 2.1.1.* Available online: http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/011/02.01.01_60/gs_NFV-IFA011v020101p.pdf (accessed on 24 January 2019). 2013.

[9] Sameer Kulkarni et al. 'Neo-NSH: Towards scalable and efficient dynamic service function chaining of elastic network functions'. In: *20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*. IEEE. 2017, pp. 308–312.

[10] Adrian Farrel et al. *BGP Control Plane for NSH SFC*. Internet-Draft draft-mackie-bess-nsh-bgp-control-plane-04. Internet Engineering Task Force, Feb. 2017. 52 pp.

[11] Paul Quinn, Uri Elzur and Carlos Pignataro. *Network Service Header (NSH)-[Review]*. RFC 8300. Jan. 2018.

[12] K Tirumaleswar Reddy et al. *Authenticated and encrypted NSH service chains*. Internet-Draft. Work in Progress. Apr. 2015.

[13] Rafael Lopez and Gabriel Lopez-Millan. *Software-Defined Networking (SDN)-based IPsec Flow Protection*. Internet-Draft draft-abad-i2nsf-sdn-ipsec-flow-protection-03. Internet Engineering Task Force, May 2017. 45 pp.

[14] Pedro R. Marques et al. *Dissemination of Flow Specification Rules*. RFC 5575. Aug. 2009.

[15] Joel M. Halpern and Carlos Pignataro. *Service Function Chaining (SFC) Architecture*. RFC 7665. Oct. 2015.

[16] Håkon Gunleifsen and Thomas Kemmerich. 'Security requirements for service function chaining isolation and encryption'. In: *IEEE 17th International Conference on Communication Technology (ICCT)*. 2017, pp. 1360–1365.

[17] Ravi Chandra et al. *Multiprotocol Extensions for BGP-4*. RFC 4760. Jan. 2007.

[18] Randy Bush and Rob Austein. *The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 1*. RFC 8210. Sept. 2017.

[19] ETSI. *Network Functions Virtualisation (NFV) NFV-MAN 001 Management and Orchestration*. Available online: http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf (accessed on 04 June 2019). 2014.

[20] ETSI. *Network Functions Virtualisation (NFV) 002 Architectural Framework v1.1.1*. Available online: http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_nfv002v010101p.pdf (accessed on 04 June 2019). 2014.

[21] Andy Bierman, Martin Björklund and Kent Watsen. *RESTConf Protocol*. RFC 8040. Jan. 2017.

[22] OASIS. *TOSCA Simple Profile for Network Functions Virtualization (NFV) Version 1.0, Committee Specification Draft 04*. Available online: http://docs.oasis-open.org/tosca/ (accessed on 05 May 2017). 2016.

[23] J Vilhuber et al. *Kerberized Internet Negotiation of Keys (KINK)*. RFC 4430. Mar. 2006.

[24] Steven Bellovin, Randy Bush and David Ward. *Security Requirements for BGP Path Validation*. RFC 7353. Aug. 2014.

[25] VPP. *Vector Packet Processing Test Framework*. Available online: https://docs.fd.io/vpp/17.04/ (accessed on 05 May 2017). 2016.

[26] Quagga. *Quagga routing suite*. Available online: http://www.quagga.net (accessed on 02 May 2017). 1999.

[27] Kenneth Raeburn. *Encryption and Checksum Specifications for Kerberos 5*. RFC 3961. Feb. 2005.

[28] Racoon. *Racoon IPSec Key Exchange System*. Available online: http://www.racoon2.wide.ad.jp (accessed on 02 May 2017). 2006.

[29] D. Perino et al. 'A programmable data plane for heterogeneous NFV platforms'. In: *IEEE Conference on Computer Communications Workshops (Infocom Workshops)*. Apr. 2016, pp. 77–82.

[30] R. S. V. Eiras, R. S. Couto and M. G. Rubinstein. 'Performance evaluation of a virtualized HTTP proxy in KVM and Docker'. In: *7th International Conference on the Network of the Future (NOF)*. Nov. 2016, pp. 1–5.

Chapter 10

Dynamic setup of IPsec VPNs in Service Function Chaining

Published in Elsevier Journal Computer Networks, 2019

Håkon Gunleifsen, Vasileios Gkioulos and Thomas Kemmerich

Department of Information Security and Communication Technology, Norwegian University of Science and Technology (NTNU), Postbox 191, 2802 Gjøvik, Norway

hakon.gunleifsen2@ntnu.no, vasileios.gkioulos@ntnu.no, thomas.kemmerich@ntnu.no

Abstract

This article describes a novel mechanism for the automated establishment of dynamic Virtual Private Networks (VPN) in the application domain of Network Function Virtualization (NFV). Each hop in an NFV Service Function Chain (SFC) lacks the capability of per-flow encryption, that makes the traffic flow in federated NFV environments vulnerable for eavesdropping. Due to the possible lack of bidirectional data plane communication channels between VNFs in an SFC, the Internet Security Key Exchange protocol (IPsec-IKE) is not applicable inside a VNF. Hence, this article introduces an alternative to IPsec-IKE that is specifically designed for NFV environments. This component is named Software Defined Security Associations (SD-SA), which is shown through a proof of concept evaluation to perform better than IPsec-IKE with respect to bandwidth and

resource consumption.

Keyworks: NFV; SFC; NSH; IPsec; IKE; SD-IKE; RESTconf

10.1 Introduction

The security mechanisms in Software Defined Networks (SDN) and NFV lack the capability to encrypt and isolate the end-user traffic between VNFs. Figure 10.1 exemplifies the problem by describing a typical VNF Service Function Chain (SFC), where the network traffic traverses multiple VNFs located at multiple service providers, while earlier work [1, 2, 3] shows that the current NFV standardization attempts from ETSI [4] and IETF [5] do not take VNF isolation into account in the SFC design.

Accordingly, an end-user who subscribes to VNF services from multiple services providers:

- Cannot end-to-end encrypt traffic, since the VNFs require to have access in order to manipulate this traffic.
- Is not aware and in control of which service providers having access to the data traffic, can potentially eavesdrop traffic and manipulate the route tables.
- Does not know if the VNFs are shared network services with other users, who can as such access private data.

Earlier work [2], [1], showed that these problems could be resolved by introducing hop-by-hop encryption per IP flow or per group of IP flows. This is enabled by deploying an encryption application in front of every VNF within an SFC (Figure: 10.1). We showed that this encryption application is typically a Virtual Machine attached to the Virtual Link [6], particularly assigned for this function. These underlying encryption functions [1] can also be perceived as regular VNFs.

Furthermore, earlier work also showed an additional problem with such an architecture. A service chain, following the NSH and SFC specification [6], can have a different service path than the reverse service path. Consequently, a pair of encrypting and decrypting VNFs in a service chain, do not necessarily have a bidirectional communication channel on the data plane, where they can exchange keys. Hence, there is a need for a new key exchange mechanism that is not dependent on a point-to-point bidirectional communication channel. This particular lack of a data plane communication channel and the need for flow-based encryption is specific to the application domain of NFV.

In this article we continue this work, focusing on the authentication and key distribution, seeking to automate the set up of secure channels between VNFs.

The investigated research problem is similar in nature to the auto-configuration of VPN setups [7], while based on the reviews of RFC 7018 [7], the earlier lack of use cases for such a protocol might be the reason for this not being resolved. Yet, the emergence of SDN and NFV technologies, highlight security use cases that necessitate renewed effort towards this direction. Accordingly, a similar problem was also stated in a recent Internet draft regarding the VNF registration process over the Interface to Network Security Functions (I2NSF) [8],[9]. The draft shows that automation of Network Security Functions such as VPNs is challenging. This is due to the lack of a secure key distribution mechanism and the lack of support for multi-vendor and multi-operator use cases (I2RS [10]). This problem is resolved by the solution presented in this article, by having a separate SDN controller handling all key distributions in a multi-operator SFC.

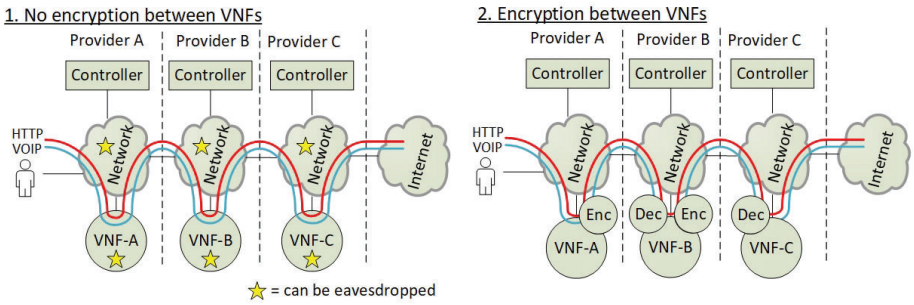


Figure 10.1: Use case and possible adversarial placement

Figure 10.2 shows how the network topology can be simplified for the use case described in Figure: 10.1, where it is assumed that one orchestration plane is capable of orchestrating the distribution of tunnel connection parameters and keys. The simplified figure shows how an IP packet from the end-user is routed through a network, with Network Service Header (NSH) [11] transport and encryption enabled per flow. Accordingly, an encrypted tunnel per flow between every VNF can ensure that end-user traffic traversing an SFC can only be accessed by the related VNFs, assuring that only these VNFs have the encryption keys to access this component of the data-flow.

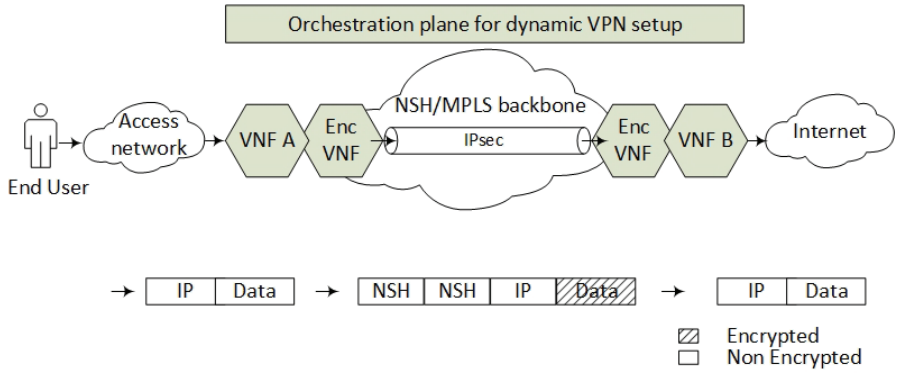


Figure 10.2: Network topology simplification

This paper introduces a mechanism for the isolation and encryption of data traffic between VNFs in a federated NFV environment. We introduce a method for mutual and secure authentication of encryption functions in an SFC

in order to establish a secure channel between them. An architecture of a site-to-site VPN setup with a new mechanism to distribute both initial keys and cipher keys is presented, while in addition to the theoretical aspect of the new protocol, the paper also presents the empirical results from experiments on the implemented design.

The main contributions in this article are:

- A set of requirements for NFV services running isolated services.
- A new architecture of a key exchange mechanism in distributed NFV environments.
- A performance and security analysis of the security mechanism proposed.

The remainder of this article is structured as follows. The most related work concerning VPN authentication follows this introduction. Section 10.2 defines the prerequisites, constraints, topology assumptions and requirements that are needed in order to apply the new authentication mechanism. The section also correlates the requirements with existing alternatives. The architecture in Section 10.3 suggests a design for automating VPN configurations, while Section 10.4 shows how this is implemented. Section 10.5 demonstrates a proof of concept experiment with performance tests. An evaluation of the performance test results is presented in Section 10.6, while Section 10.7 concludes this paper. NFV allows Internet Service Providers to provide flexible network service deployments. However, recent research [12] have shown that this new technology should be secured from both the

service provider and the end-user perspective. NVF surveys [13], [14], [15] have pointed out multiple threats and vulnerabilities in NFV, where end-user privacy is an open issue. Due to the fact that VNFs are acting as middleboxes and require access to the data-content, makes end-to-end encryption difficult. Multi-Context TLS [16] was developed to solve this issue. However, the protocol is insecure [17] and it provides neither flow-based encryption nor SFC isolation. Hence, another approach to this problem is to enable the NFV infrastructure to provide hop by hop encryption and automatically exchange keys and set up secure channels between the VNFs. Our previous work [3] outlined the top-level architecture of such an authentication and key distribution protocol, and was motivated by the lack of security features in the architectural guidelines from ETSI standardizations, IETF, and academic research [3], since no protocol was found supporting the (1) authentication of VNFs, (2) negotiation of keys and (3) dynamic setup of secure VPN connections between VNFs. However, the principles of such requirements are similar to the Generic Bootstrapping Architecture (from 3GPP) [18] and Kerberos [19]. The main difference to this research problem is that it is the cryptoVNF and not the end-user that is involved in the authentication process.

In networks controlled and provisioned by operators, it is common to use a management plane or a control plane to provision the network topology. Therefore it is also possible to run the authentication process in a separate control plane domain. This is similar to the separation of the control and data plane in SDN where SD-IKE [20] is commonly utilised. However, SD-IKE does not describe how to securely distribute the keys between the network controller and the VNF, neither how the VNFs can preserve SFC packet headers during encryption, nor how it performs compared to regular IPsec setups.

This lack of SFC header preservation is also reflected through our previous work [2], showing that in contrast to the data, the SFC headers cannot be encrypted in order to enable the routers to perform SFC routing of the encrypted data. Therefore, the SFC header must be located between layer 2 and layer 3. Hence, encryption of layer 2.5 or layer 3 is needed in order to not interfere with the end-user data, where layer 3 encryption by the use of IPsec [21] in transport mode is preferred. IKE [22] and IKEv2 [23] are the main protocols used for key negotiations for IPsec, but currently, they require modifications in order to support a dynamic NFV environment with IKE over NSH.

Due to the backbone network in the examined scenarios (Figure: 10.2), encryption should ultimately be performed on the NSH layer. However, encryption on the NSH layer has currently no standard, except for securing the integrity of the NSH headers [24]. Furthermore, encryption on the upper transport layer by the use of SSL/TLS such as OpenVPN [25] is also possible. However, the end-to-end

transport layer between VNFs is distinct from the end-user transport layer, creating additional overhead and potential packet segmentation or delay. Also, TLS based tunnelling is often based on endpoint attributes, such as an URL identity, something that does not fit to a site-to-site VPN setup. Similarly, the Wireguard [26] protocol that is an alternative to OpenVPN and IPsec, also simply encapsulates encrypted packets in a UDP header. Yet, Wireguard and OpenVPN have no good solution for key distribution and key derivation, as one key pair is used in the long-term and for all communication [27].

Furthermore, IETF proposed an Interface to Network Security Function (I2NSF) [28], to enable the exchange of secured messages between the VNFs and a security controller. This approach focuses on an out-of-band interface to operate the VNF, but this interface lacks security functions for authenticating and validating the VNFs. Hence, another key feature of our adversary model, is the characteristics of a separate control and management plane in NFV. This article adopts elements from authentication and key distribution in related autonomous control plane backbone networks [29], [30] to an NFV environment. Also, principles from distributed authentication protocols in Machine-to-Machine networks [31] and ad-hoc networks [32] [33] can be adapted to the NFV domain. Accordingly, the requirements both for authentication and key negotiation are defined in Section 10.2, in order to classify how the existing authentication protocols match the corresponding operational requirements.

Other related studies can also be identified in the literature, including Dynamic VPN architectures [34], Distributed VPN systems over peer-to-peer networks [35] [36] and security protocols for distributed systems [37]. All these articles refer to similar problems, but within distinct application domains, and although relevant, the proposed solutions are not directly applicable to the specifics of federated NFV environments.

10.2 Extraction and discussion of requirements

The core requirement for the examined scenario is a centralised system that can dynamically configure pairwise VPN channels between VNFs. IPsec is the most common approach for supporting encryption, wherein other environments, the two parties have a preshared key or a set of PKIs that is used to negotiate an encryption key by the use of the IKE protocol. IPsec is also selected as the encryption protocol for NFV. Yet, a mechanism is required in order to support the dynamic setup of IPsec channels, since the nature of VNFs in an NFV environment differs from normal endpoints in an IPsec channel. These differences are based on the following constraints of the NFV environment:

- It is the encrypting service function and not the end-user that act as VPN clients. Hence, the VPN setup is perceived as a site-to-site VPN setup, where the sites (or gateways in this case) are VNFs with encryption capabilities. From an end-user perspective, it should be of no concern how the dynamic VPN is provided, because neither the end-user nor the end-user device are participating in the process.
- The VNFs dynamically change their connection topology according to the specified SFC. At one point in time, VNF A is connected to VNF B, while at another point in time VNF A is connected to VNF C (Figure: 10.3). Both a network failure or a user-initiated request can trigger such a change in the service chain, at which time a centralised service such as an Authentication Centre (AuC) must inform the VPN gateways to reconfigure the VPN channels and derive new encryption keys.

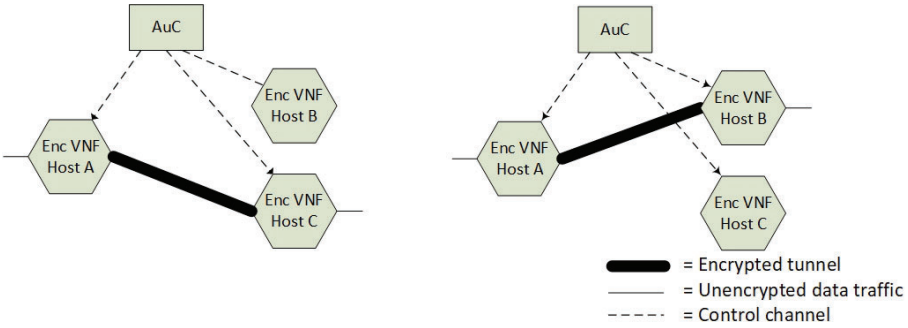


Figure 10.3: The dynamic behaviour of VPN tunnels

- It is the orchestration system (OSS) that defines the service chain and therefore controls the topology. Hence, the OSS must distribute VPN tunnel properties to the VNFs (which are acting as VPN peers). Therefore, the setup of the VPN peers must be initiated from a centralised unit such as an SDN controller or the OSS.
- In NFV environments, VNFs are enabled by containers or virtual machines, which operationally have a slow startup (especially a virtual machine). Therefore, it is not preferred to boot up VNFs on demand every time a service chain changes.
  Hence, the encryption functions must be pre-initiated for every compute node that contributes in an SFC. Also, the encrypting service function must have a secure channel to the Authentication Centre (AuC).
- A VNF application running encryption does not have to consider routing. The underlying NSH/MPLS layer is ensuring that the packet is routed correctly. Hence, the VNF encryption application can route all traffic through the ap-

plication.

These constraints and functional requirements result in a set of requirements specific to the processes of authentication and key negotiation.

- **A trusted third party:** The involved parties must be authenticated (VNF-VNF and VNF-AUC), preferably with the use of a dedicated authentication server.
- The authentication and connection properties are dependent on the SFC. Hence, **context-based authentication** is required, in order to determine the physical location and SFC belonging of the VNFs.
- The AuC must be able to authorise service requests after authentication. This implies **distinguishing and isolating service requests** from different clients. For example, if a session between the VNFs and the AuC ensures confidentiality for transporting the service request, the AuC application also has to associate the incoming service request with this session. This ensures that a VNF cannot inject service requests for another VNF.
- Due to the SFC transport between VNFs, a dedicated control channel, typically provided by the backbone network, must be used for the authentication protocol. This control channel is preferable since the setup of VPN tunnels and backup tunnels should not be dependent on the NSH/MPLS tunnel, while this communication must be **carried by the IP**.
- It is the VNF that has to initiate the authentication process since it must announce its presence. However, it is the AuC that must initiate the setup of an IPsec channel between two VNFs, since the AuC is the party that knows which VNFs that are required by the SFC. For that reason, the authentication protocol needs to support **both server-side and client-side initiated authentication**.
- Each VNF must be uniquely identified, with the identifier being pre-provisioned from the orchestration system to both the AuC and the VNF. **The VNF identifier** and the key are considered equivalent to a username and password pair.
- Due to the dynamic topology of the VPN channels, the corresponding keys cannot be static. This implies that in addition to the pre-provisioned static keys, additional keys for tunnel setup between the VNFs must be dynamically derived. After an initial authentication between the VNF and the AuC, these derived keys must be transferred securely to the VNFs. However, the initial keys used in the initial authentication must also be protected from eavesdropping. Hence, tickets or random numbers are needed in order to protect these credentials with the use of **confidentiality mechanisms during key derivation.**
- When a VNF is authenticated by a third party, the protocol must supply the

VNF with a **remote connection gateway** for VPN setup.

Table 10.1 maps these requirements to the properties and supported functions of existing authentication protocols.

	Third party auth	Context-based auth.	Service request isol.	Supports IP transp.	Serverside auth	VNF identifier	Key derivation conf.	Remote connection attr.	
PPP protocols	✗	✗	✗	✓	✗	✓	✗	✗	PAP[38], CHAP[39], MSCHAP[40]
AAA protocols	✗	✓	✗	✓	✗	✓	✗	✗	RADIUS[41], TACACS[42], DIAMETER[43]
Protocol overlays	✗	✓	✓	✓	✓	✓	✓	✗	EAP[44], PEAP[45], PANA[45], LEAP[37]
IP layer	✗	✗	✓	✓	✓	✗	✗	✗	IKE-P1[22], IKEV2-AUTH[23]
Transport and session layer	✗	✗	✓	✓	✓	✗	✗	✗	TLS[46][47], DTLS[48]
Security applications	✓	✗	✓	✓	✗	✓	✗	✗	SAML [49], OAuth [50]
Generic bootstrapping	✓	✓	✓	✓	✓	✗	✗	✗	EAP-AKA[18]
Orchestration appl.	✓	✓	✓	✓	✓	✓	✗	✗	DMVPN[51], EAP-KMS[52], GSAKMP[53]
SW Security frameworks	✗	✗	✗	✗	✗	✗	✗	✗	SASL [54], GSS-API [55]
Key Management Systems	✓	✓	✓	✓	✗	✓	✓	✗	KERBEROS[19], KINK[56]

Table 10.1: Authentication protocol

Point to Point Protocols (PPP) such as PAP [38], CHAP [39] and MSCHAP [40] are mainly designed for link layer authentication and do not focus on key distribution from a third party. These methods are also often used in AAA protocols such as RADIUS[41] and TACACS[42], reflecting point-to-point client-server authentication. Protocol overlay frameworks, such as EAP[44], have been developed to use an underlying protocol to carry the EAP messages. This is also typically designed to be used when IP is not available and there is a need for carrying authentication messages over multiple link-layer hops. An authentication method connected to layer 3 or layer 2.5 encryption is the main objective of this article. IPsec authentication variants have different methods for authenticating two peers, such as IKEv1 [22], IKEv2 [23] or KINK [56]. However, they all assume that the peers know the address of a remote peer before the authentication method starts, accordingly necessitating their extension in order to fully satisfy the aforementioned requirements.

Hop-by-hop tunnelling can also be achieved by enabling tunnelling on the transport or the session layer (layer 4/5). However, these hop-by-hop tunnels must not be mixed layer 4-7 data from the end-user such as SSL/TLS [47] layers. This implies that such tunnels must be implemented as underlying hierarchical tunnels where IP is transported over a layer 4-7 tunnel, which would require additional underlying hop-by-hop IP tunnels and an overlying orchestrating application. This additional overhead makes such tunnels non-preferable. Furthermore, application-based authentication relies on authentication messages that are encapsulated by application plane markup languages such by XML or REST messages. Examples of such protocols are OpenID and SAML, while these messages often rely on an end-to-end transport mechanism, such as TLS, in order to ensure the confidentiality and integrity of the messages. This does not resolve the underlying identification and authentication problem where the VNFs do not know the remote endpoint and a URL does not exist.

Another approach for orchestrating authentication is to distribute network configuration through a network orchestrator, which is a common approach for many network vendors. Cisco uses for instance Group Encrypted Transport (GET) [57] to distribute VPN configurations from a server down to the clients, with multiple similar protocols and standards been suggested for the distribution of such configurations [58],[51], [59]. Yet, these solutions only distribute the initial keys, without having the capability of changing the configuration of the VPN topology rapidly and dynamically. Furthermore, other IPsec extensions distribute keys more efficiently [53],[60], but they can neither distribute information about the endpoints that need to be connected, nor rely on a third party being responsible for endpoint configuration.

The dynamic key distribution is one of the most critical features in designing automation of VPN setup for VNFs. A relevant approach is to use a dynamic key distribution protocol such as Key Management Systems that includes two-sided authentication such as Kerberos [19] or GPAKE [61]. However, Kerberos has security properties which impede the orchestration in a multi-domain environment [62], especially in cases where the remote VPN peer must be received dynamically. Also, GPAKE has similar restrictions and does not have the capability of uniquely identifying a VNF identifier. On the other hand, Kerberos has an IPsec authentication extension named KINK [56] that makes it suitable for combining it with IPsec with service-side authentication. However, the protocol only supports IKEv1 and it has no distribution of remote endpoints. Accordingly, no existing protocol was found to fully satisfy the aforementioned requirements. Yet it was identified that a suitable solution would be a framework around IPsec, that also enables a fast, dynamic, and flexible key distribution. Furthermore, from the perspective of an NFV

operator, an orchestrated solution fitting into the NFV framework is preferable, such as an API based architecture with principles from RESTconf [63].

10.3 Architecture

Based on these results, an authentication protocol and a key distribution mechanism are suggested. Figure 10.4 explains the top-level operation of the protocol derived from the aforementioned constraints, within a simplified scenario with 2 VNFs and an Authentication Centre (AuC). The simplified process consists of VNF instantiation, VNF authentication and VNF Configuration.

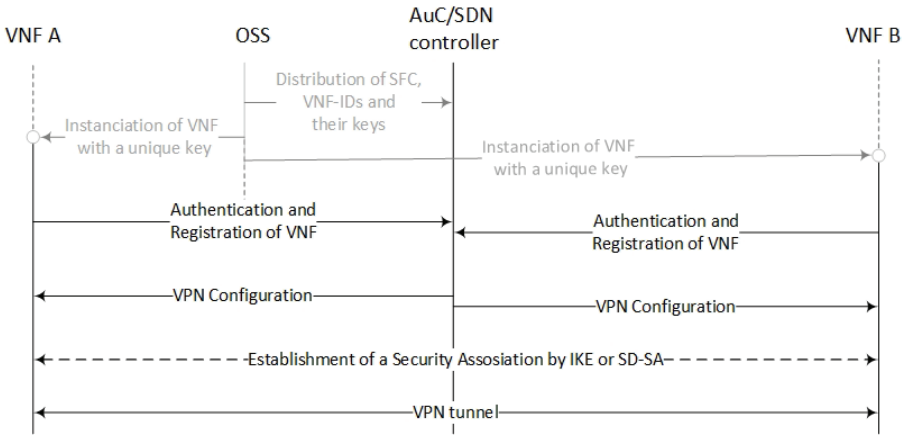


Figure 10.4: Simplified operation

We have based the proposed architecture on the principles of IPsec and RESTconf, building the framework around RESTconf in order to enable it to configure IPsec running inside VNFs in a dynamic manner. Our main contribution is therefore an architecture that automates the setup of IPsec over RESTconf where the IPsec services are running inside a VNF in an NFV environment. The proposed framework consists of three main components: (1) The VNF with encryption capabilities (VNF aka cryptoVNF), (2) an Authentication centre (AuC) and (3) an SDN controller, while all the components communicate by web services using the JSON format [63]. Figure 10.5 shows the bootstrap sequence and the communication between the different components. The bootstrapped mechanism consists of five steps:

0 Creation of VNFs, pre-distribution of keys, and definition of an SFC.
1 Mutual Authentication between the VNFs and the AuC.
2 Setting up configuration channels (RESTconf) between the VNF and the

AuC.

3 Distribution of VPN configuration to the VNFs

4 Tunnel setup Local application.

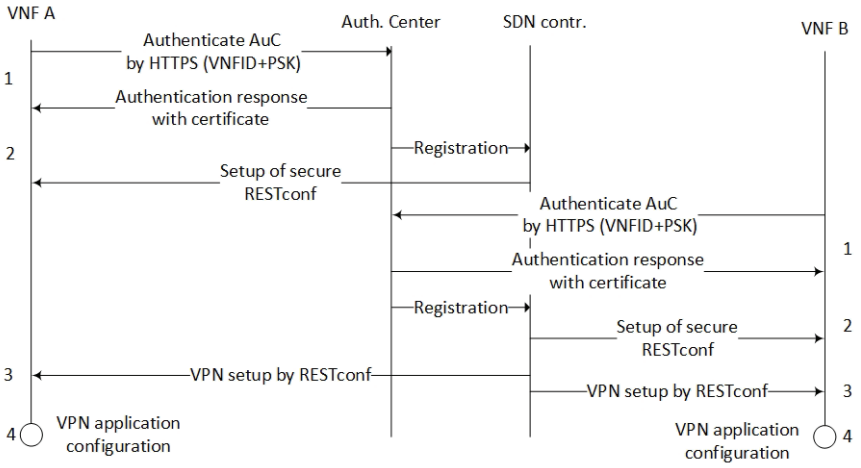


Figure 10.5: Detailed sequence diagram

0 **Distribution of VNFs with encryption capabilities.** We assume that an orchestration system is requested to set up a service chain and accordingly calculates the number of VNFs and encrypted channels. The VNF manager could then instantiate the VNFs, and pass them a globally unique VNF identifier and a preshared key. In a VMware environment, these can be set as parameters in the VMX file, enabling the guest OS to retrieve this information from the hypervisor during bootup. We skipped this initial step from the implementation of the architecture (numbered as 0), since the orchestration system functionality is not the focus of our contribution, while for proof of concept purposes the identifier and key were manually encoded into the machine.id parameter in the VMX file together with the hardcoded address of the AuC.

1 **Authentication and registration of an encryption VNF service.** After the VNF Manager has booted the VNF services, a registration service is initiated for the VNF, collecting the VNF identifier and the preshared key from the VMX file. Accordingly, the same service authenticates itself towards an AuC web service, where the authentication is ensured by SSL (HTTPS) with a certificate connected with the domain name. After a successful registration, the AuC pushes the identifier and the IP address of the VNF to a database of authenticated VNFs running on the SDN controller, while for every DHCP change the VNF reauthenticates.

2 **Setting up RESTconf** The second phase in the initialisation process is to establish a secure connection from the SDN controller to the VNF, in order to enable secure RESTconf messages. Because of the process in step 1, the SDN controller now has the IP address and the certificate to establish the TLS enabled RESTconf connection.
3 **RESTconf configuration pushing.** An SDN controller application is configured to push VPN configuration down the VNFs when all VNFs have booted and registered. The southbound interface of the SDN controller uses RESTconf to send a set of standardised configuration settings defined by NETconf YANG [64].
4 **IPsec application setup** The last step in the process is to configure the VPN application, which parses the NETconf YANG configuration into the application specific configuration settings.

This mechanism also allows the encryption application running inside the VNF to be unaware of the remote VPN peer when it is instantiated. Furthermore, the proposed mechanism also enables the use of Software Defined IKE (SD-IKE), because the two peers in the VPN are already authenticated towards an SDN controller, IPSec IKE becomes redundant. Accordingly, it is possible for the SDN controller to distribute the keys instead of IKE negotiations. Hence our second contribution in this paper is an architecture that enables automation of the setup of SD-IKE. The current SD-IKE draft suggests to provision this over I2NSF, which has limited features in a multivendor setup. Therefore, here we present an architecture for automating the VNF setup by the use of RESTconf and standard VNFs. We call the proposed approach Software Defined Security Associations (SD-SA), which is based on provisioning standard IPsec without IKE but over RESTconf.

Figures 10.6 and 10.7 show the differences between running IPsec with (existing approach) and without (suggested approach) the IKE protocol. The main objective of the IKE protocol is to authenticate the peers in order to populate the Peer Authorization Database (PAD) and distribute symmetric keys by populating the Security Policy Database (SPD) and Security Association Database (SAD). Within the mechanism presented earlier (Figure 10.5), the peers are already authenticated and the centralised controller is capable of replacing the IKE protocol. Instead of requiring IKE to populate the kernel databases, the SDN controller is populating directly the SAD and SPD databases, in order to reduce IKE resource consumption on the peers. In normal IKE setups, a Security Association is established between the peers, where the keys are transmitted. When not running IKE, the SA is not established between the peers, but the keys are distributed by the SDN controller. Since a secure channel exists for VPNGW1-SDNContr and VPNGW2-SDNContr (Figure: 10.7), the sum of these two channels is perceived as the aforementioned

Software Defined Security Association (SD-SA).

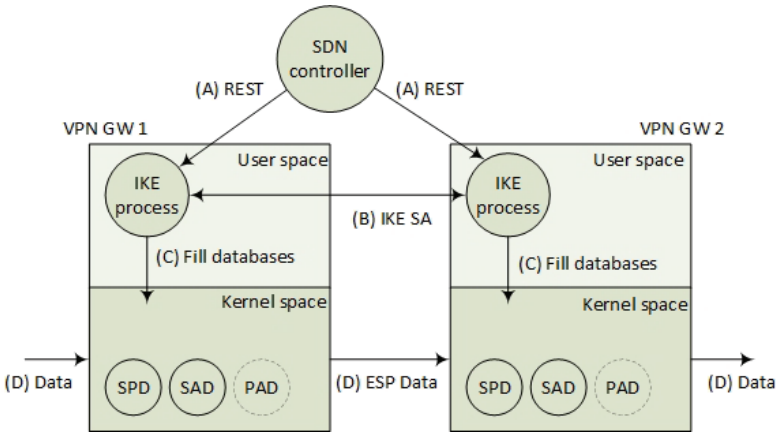


Figure 10.6: Updating IPsec configuration by RESTconf and making a Security Association (SA) by using IKE

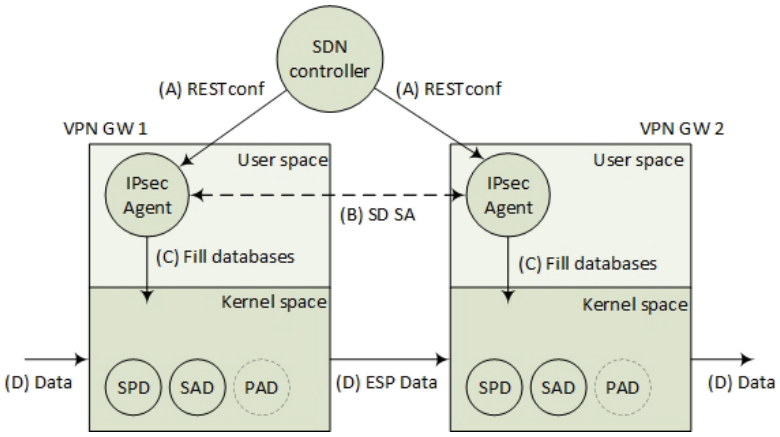


Figure 10.7: Distributing keys directly from controller by RESTconf and making a Software Defined Security Association (SD-SA)

In these two cases, the RESTconf API must support the distribution of 1 - RESTconf based IKE configuration and 2 - distribution of PAD+SAD configurations of RESTconf (SD-SA). In the first case, a REST message is sent with basic IKE preshared keys and connection settings, while in the second case the REST messages contain the same symmetric integrity key and encryption key.

10.4 Implementation

The main objectives of the implementation were to present an instance of the proposed mechanisms, to perform a proof of concept test and a performance comparison between IPsec/IKE and IPsec/SD-SA. Figure 10.8 shows the four main components in the software design:

1. The registration service on the VPN peer
2. The SDN controller acting as a configuration client and AuC
3. The local configuration service on the VPN peer
4. The IPsec service

Furthermore, Figure 10.9 provides a simplified flowchart and pseudo-code based description of the required processes.

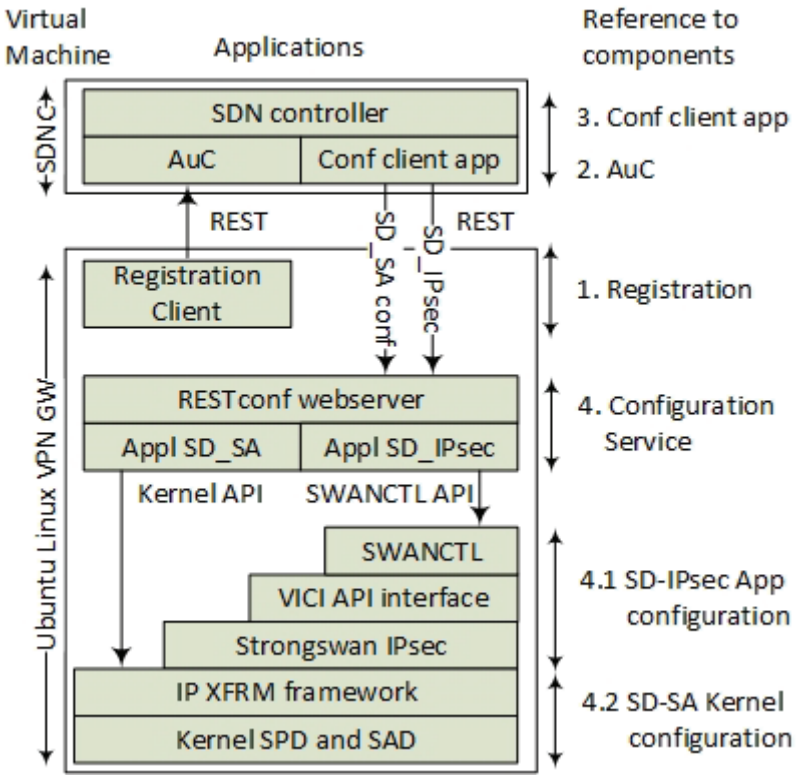


Figure 10.8: Components in the architecture

0 **The VNFManager** is omitted for the implementation. That means that we manually defined the VPN pairs in the AuC. We also created an ISO image template in VMware and hardcoded the VNF-ID and the preshared key

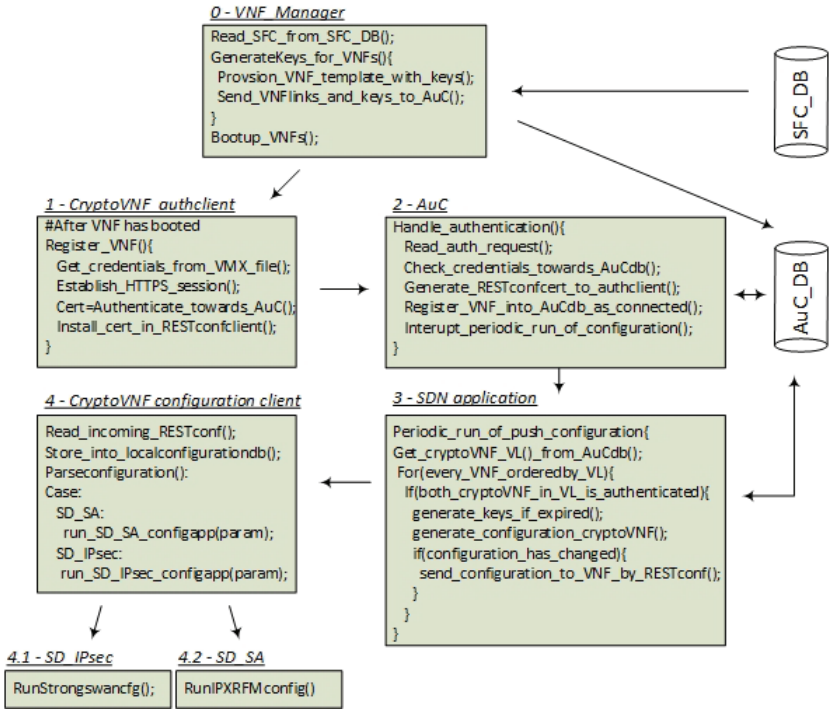


Figure 10.9: Flow chart and pseudo-code description of the processes

during the instantiating of the VNFs (in this case virtual machines).

1 **The registration client** is running in the VPN peer (the VNF) ensuring that a secure channel is set up between the controller and the VPN peer. On the VPN peer, the guest operating system application package OpenVMtools provides an API to get the VNF-ID and the PSK from the hypervisor. The application sends these credentials as an HTTP request to the AuC. This was implemented as a Linux bash shell script running wget commands. The service also updates the controller with the management address of the VPN peer.

2 **The Authentication centre** is implemented as a web service running one authentication service. In the proof of concept experiment, the database is implemented as a simple text file containing all registered VNFs, their IDs, their PSKs and their management IP addresses. The web service authenticates the registration client by its ID and PSK, while the registration client authenticates the AuC by a certificate over a standard HTTPS connection.

3 **The Configuration client** runs a script that reads two sets of text file databases. The authenticated VPN peers and the set of defined VPN pairs. When

a set of VPN peers is defined and both peers are registered, it sends the corresponding VPN configurations to the VPN peers. The configuration application reads the databases periodically and updates the VPN peers (the VNFs) with their configuration by RESTconf. Two versions of configuration options are enabled. The SD-IKE configuration configures IKE (Figure: 10.10), while the SD-SA configuration sets up an SA without the use of IKE (Figure: 10.11). The format of the configuration is based on the experimental updates of the expired IETF IPsec YANG specification draft [64]. For experimental purposes, we used a pseudo-JSON yang format that only contained a subset of the most important configuration parameters. For SD-IKEc the remote gateway, the identifier, the key lifetime and a preshared key were the most important parameters for making our proof of concept. For the SD-SA we defined 4 basic SDP policies per connection containing the connection id (ReqID) and the relevant IP address endpoints. Correspondingly, the SD-SA SAD state configuration contained only the relevant IP addresses of the endpoints, the IPsec header tags (SPI), the connection IDs (ReqID), the integrity keys and the encryption keys.

```
{"ipsec":
 {"ikev2":{"ike-connection":{"ike-conn-entries":
  {"conn-name": "netnet_1",
   "phase1-lifetime": "3600",
   "phase2-lifetime": "4",
   "local":
    {"ipv4":"192.168.161.2",
     "my-identifier":"VNFID:AS1-111"},
   "remote":
    {"ipv4":"192.168.162.3",
     "my-identifier":"VNFID:AS2-222"}
.....
  "secrets":
  {"secret":
   {"id":"VNFID:AS1-111",
    "psk:"psk1234"}}
  {"secret":
   {"id":"VNFID:AS2-222",
    "psk":"psk4321"}}
....
}}}
```

Figure 10.10: RESTconf YANG JSON data IPsec configuration (Step 3 to 4)

4 **The configuration service** is the local configuration server that is running on each VPN peer (VNF). The main objective for this service is to receive

```
{"ipsec":
 {"spd":{"spd-entry":{"condition":{"trafficselectorlist":
    {"direction": "OUTBOUND",
     "local-address":
      {"start":"192.168.162.1",
       "end":"192.168.162.1"}
.....
 "sad-entry":
 {"spi":"12345"
.....
   "esp-sa":
   {"encryption":
    {"encryption-alg":"aes",
     {"key": {"encr_key", "iv"="vector"}
   {"integrity":
    {"integrity-alg": "hmac-md5-128",
     "key": "int_key"}
....
}}}
```

Figure 10.11: RESTconf YANG JSON data SD-SA configuration (Step 3 to 4)

the VPN configuration from the centralised configuration server. This is implemented as an SDN controller that stores the VNF specific configuration locally. The application is storing the incoming configuration in XML files based on pseudo-RESTconf/YANG. We also simulated the notification service in NETconf by letting the service request trigger on-demand configuration changes to increase the deployment speed of the configuration. That was reflected by two different configuration web services that update the configuration. One application (4.1) configures the IPsec application with IKE configuration (SD-IKE) and another application (4.2) updates the kernel directly with the IPsec keys (SD-SA). Figure 10.13 and Figure 10.12 shows examples of the two southbound scripts that are configuring IPsec.

4.1-4.2 **IPsec application** - We used Strongswan as the IPsec IKE application. The advantages with this application are that it supports both manual IKE configuration in text files, and that it has an API for controlling the IPsec configuration on demand. The Strongswan IPsec application has a dynamic library that enables such an interface. However, we utilised the swanctl application overlay in order to enable Linux bash scripting CLI commands for updating the IPsec configuration without restarting the service. For the SD-SA application that manipulates the kernel IPsec configuration, we utilised the IP XFRM framework (Figure 10.13).

```
#script example manipulating IPsec configuration

#!/bin/bash
./generateconfigfromYang.sh > /usr/local/etc/ipsec.conf
ipsec rereadall;
for (( f=0; f<=$ConNb; f++ ))
do
 changeconfig = ´./getconfigstatus.sh $ConNb´
 if [ $ConNb -eq changeconfig ]
 then
   swanctl --terminate --ike $connections;
   swanctl --initiate --child ConNb$connections
 fi
done
```

Figure 10.12: Code example IKE configuration application (Step 4 to 4.1)

```
#Script example updating IPsec in the kernel with IP XFRM

ip xfrm state add src $IPsrc dst $IPdst proto esp spi 0x53fa0fdd
mode transport mark 0x$LblMark reqid $ReqID replay-window 32
auth "hmac(sha1)" 0x$Key1 enc "cbc(aes)" 0x$Key2

ip xfrm policy add dir out mark 0x$LblMark src $rcNet dst $Dstnet
ptype main action allow priority 2080 tmpl src $IPdst dst $IPsrc
proto esp reqid $ReqID mode transport

.....
```

Figure 10.13: Code example IP XFRM config application (Step 4 to 4.2)

10.5 Verification by experiments

The implementation was tested in order to run a proof of concept for the authentication and IPsec deployment mechanism, and to make a comparison of performance between IPsec/SD-IKE and IPsec/SD-SA. The performance test is primarily conducted in order to measure how much time it takes to change the data plane keys and how much overhead the key exchange protocol introduces with respect to resource consumption. Changing the data plane keys introduces a packet loss during rekeying. In IKE version 2, the specification states that a new child SA should be established before the existing child SA is deleted. However, our measurements

show that, even in this case, the Strongswan implementation still loses packets during rekeying. By sending a fixed stream of UDP packets through the VPN tunnel, the time-gap between rekeying the two peers is calculated (Formula: 10.1).

$$\text{Time gap} = \frac{\text{packets lost per key-change}}{\text{packets sent per second}} \tag{10.1}$$

In IKEv2 rekeying contains two major components. These are 1- Rekeying and reauthentication of the IKE SA session (aka Phase1 in IKEv1) and 2- Rekeying of the child SA (aka Phase2 in IKEv1) that contains the encryption and integrity key for the data plane. Usually, IKE SA rekeying is initiated every 3 hours, while child SA once every hour. In SD-SA, the IKE session is not established, while the rekeying is pushed by configuration changes from the controller. Hence, both IKE SA and child SA rekeying is compared with rekeying of SD-SA. Additionally, a key element in the SD-SA design is to compare the reestablishment of IKE. The requirement of the dynamic behaviour of reconnecting IPsec VNFs during a service chain modification (Figure: 10.3) is a feature that IKE is not designed to handle. However, SD-SA does not handle such configuration differences differently than normal rekeying. In our experiment, we have omitted routing table changes in such setups and we have also assumed that routing and key distribution is performed in one operation. Also, all IKE experiments are performed with the IKE version 2 since this version is known to be faster than IKE version 1 [65].

The requirement section (Section: 10.2) stated that the encryption functions are considered being pre-instanciated. In an NFV context, this implies the encrypting functions are available from a pool of specifically assigned and pre-instanciated VNFs outside of the regular VNF application domain. The deployment time of using an encryption function is therefore considered as the time it takes to establish a new VPN configuration, by sending a message from the AuC to the CrytoVNF. Hence, we do not measure deployment time and failover time, but we perceive the equivalent as IKE reconnection and SD-SA reconnection times.

Based on these IKE attributes we created five test scenarios in our experiment. Three IKE scenarios and two SD-SA scenarios.

- A reference performance test for IKEv2 is defined when no key-change takes place. This ensures that there is no packet loss without key-changes and defines the maximum bandwidth throughput
- A reference performance test for SD-SA was also defined with no key-change. This test case also verifies that there is no packet loss without key-changes.
- An SD-SA proof of concept implementation was tested to measure the resource consumption, the delay and packet loss during key-changes, and a bandwidth test measure the overall performance of the system.

- Running IKEv2 with child SA key-change is defined to measure and compare the SD-SA with the regular IKE key-changes.
- Running IKEv2 with IKE SA key-changes is defined to measure and compare these types of key-changes to SD-SA.
- The last test was IKE reconnections, to simulate VNF changes and compare it SD-SA reconnections.

As mentioned earlier, IKE normally uses 3600 seconds as a default interval value for rekeying. In order to calculate the time gap between the host during key-changes, we ran the tests with high key-change rates. These tests are not relevant in normal IPsec setups, but allow capturing the required measurements that ensure reliable results. We ran tests with 2- and 4-second rekeying interval for packet loss measuring, while for resource consumption measurements we used 4 and 30 seconds rekeying interval.

The test was performed in a VMware lab environment provided by Eidsiva broadband in Norway. 6 ESXi host based on HP Proliant DL360G9 with 24 CPUs x 2.6 GHz and 8 Gigabit Ethernet ports. 6 virtual machines were created, one per ESXi host. Each virtual machine was allocated 4 virtual CPUs and 8 GB of RAM. All hosts were installed with Ubuntu server 16.04 LTS and were running kernel 4.4.0-116 SMP.

All servers were installed with standard installation settings with no kernel modifications.

In order to reduce the number of unknown variables in the virtual machines, no additional services were installed. Additionally, the resources were reserved to the virtual machines and no other virtual machines were running on each ESXi host to ensure no resource sharing.

Figure 10.14 shows how two test agents were transmitting packets through a site-to-site VPN topology. The router in the middle had no other purpose than ensuring that non-encrypted traffic was able to pass the router. Each Virtual machine was interconnected with dedicated network 1 Gbps Ethernet interfaces, while the connections to the SDN controller were separated from the data plane interfaces.

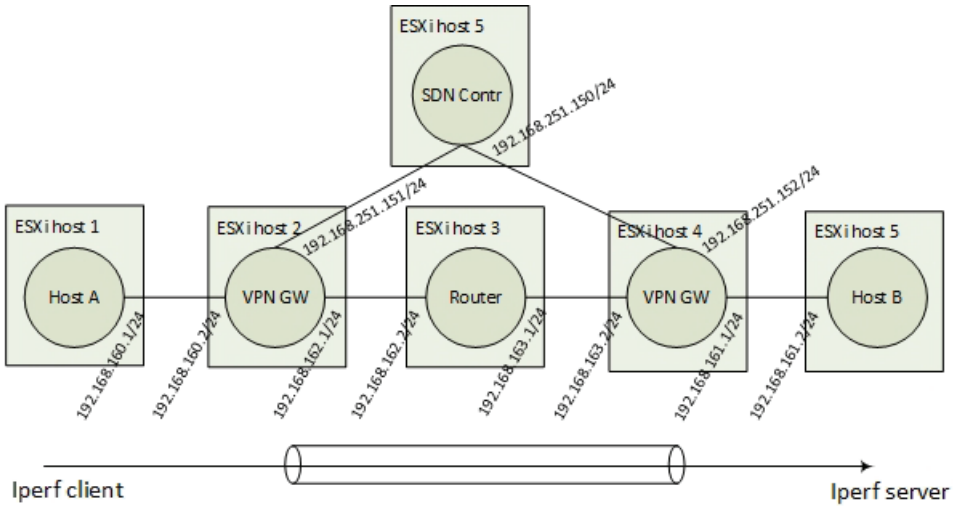


Figure 10.14: Lab topology

The VPN gateways were running Strongswan 5.5.0 git commit 8eea280, while the test agents where running iperf 5.0.2. The web services were simulated by the use of the socat application. This application is similar to netcat and capable of creating TCP/IP sockets with SSL support, and was used to send pseudo REST commands based on the Linux bash scripting that was mentioned in the implementation section 10.4.

We used iperf as the testing tool for packet loss and bandwidth tests, performing bandwidth tests with a window size of 416K in 20 seconds. For UDP packet loss tests we ran a UDP stream of 100Mbit with packet size 578 bytes at 173100 pps. Additionally, a 50Mbit test with packet size 578 at 85072 pps was performed. Each test was performed 10 times where the result is an average this. For CPU resource consumption we took periodic measurements of the CPU usage by the nmon tool, with a total of 60 samples in discrete time. Finally, to measure the memory consumption we used the fstab tool to perform 60 discrete time measurements of the memory consumption. The results of the measurements are shown in Table: 10.2 and Table: 10.3.

Most of the measurements presented in tables 10.2 and 10.3 are statistical averages, where variables such as packet loss and CPU number of software interrupts can potentially affect the outcome. Yet, the results were verified by completing the test two times with a resulting variance of no more than 5%.

		Settings		Bandwidth	100 Mbit UDP				50 Mbit UDP				Timegap
		Key change rate in seconds	Number of connections	TCP bandwidth Mbps Windowsize 416K Average of 10 tests	Average packets lost of 173100 pps 10 tests x 20 seconds	Average packetloss per key-change	Average timegap in ms per key-change	Average packets lost of 85072 pps 10 test x 20 seconds	Average packetloss per key-change	Average timegap in ms per key-change		Average timegap in ms per key-change
SDSA	Reference test no keych.	3600	1	717,4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
SDSA	SD-SA with key-change	2	1	660,0	828,0	82,8	3,9	176,1	17,6	4,1		
SDSA	SD-SA with key-change	4	1	702,0	348,8	34,9	4,1	162,1	32,4	3,8	4,0	4,0
IKEv2 SAs	Reference test no keych.	3600	1	701,8	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
IKEv2 SAs	Child SA key-change	2	1	655,8	166,1	16,6	2,1	48,2	4,8	1,1		
IKEv2 SAs	Child SA key-change	4	1	682,0	67,4	13,5	1,8	29,4	5,9	1,5	1,6	1,6
IKEv2 SAs	IKE SA key-change	2	1	NA	NA	NA	NA	NA	NA	NA		
IKEv2 SAs	IKE SA key-change	4	1	651,0	157,5	31,5	4,1	58,7	11,7	3,1	3,6	3,6
IKEv2 SAs	IKE reconnect	2	1	569,0	10878,0	1088,8	127,4	4633,0	463,3	108,8		
IKEv2 SAs	IKE reconnect	4	1	609,0	4948,0	989,6	115,8	2388,0	477,6	112,2	116,5	116,5

Table 10.2: Measuring performance and packet loss per key-change

194

Table 10.3: Measuring scalability and resource consumption

	Settings		Bandwidth	CPU		Kernel memory use		User memory use	
	Key-change rate in seconds	Number of connections	TCP bandwidth Mbps Windowsize 416K Average of 10 tests	Average CPU kernel use, 4 of 4 CPUs in 20 discrete time measures	Average CPU userspace use, 4 of 4 CPUs 20 discrete time measures	Average of total kernel memory use in bytes, 20 discrete time measures	Kernel memory use per connection in bytes	Average of total user memory use in kbytes, 20 discrete time measures	User memory use per connection in kbytes
SD SA	4	30	699	2,60%	1,00%	129k	4,3k	0	0
	30	1000	648	5,80%	2,50%	4361k	4,4k	0	0
IKEv2 w/child SA key-change	4	30	668	1,60%	1,20%	125k	7,2k	13502k	453k
	30	1000	610	4,30%	3,20%	7817k	7,8k	38284k	38k
IKEv2 w/IKE reconnect	4	30	613	1,10%	2,60%	144k	4,8k	48153k	160k
	30	1000	603	22,50%	31,00%	7603k	7,6k	62705k	63k

10.6 Discussion and Evaluation

The implementation and tests of SD-SA verify that it is possible to run IPsec without IKE in software-defined environments such as NFV. Instead of exchanging keys directly between two peers, a third party configuration server can distribute the keys directly to the peers. This is archived through pairs of secure configuration channels that are established between the VPN peers and a configuration server and replace the IKE channel. Our proof of concept implementation has the RESTconf configuration interface both can set up IKE configurations and SD-SA configurations. Accordingly, the proposed mechanism satisfies the requirements for efficient and dynamic configuration of IPsec VPNs in NFV environments. In this section, we discuss our performance results, analyse the security of our new mechanism and evaluate interoperability concerns.

10.6.1 The performance results

Tables 10.2 and 10.3 showed the result of our performance test. Here, we discuss these result with respect to key-changes, throughput, resource consumption, latency and general benefits of SDN.

IKE SA key-change

A normal IKEv2 SA key-change (Phase 1 in IKEv1) runs periodically every 3-4 hours. However, this experimental setup with short key-change intervals, intends to compare the performance of the different IPsec key-changing methods. The results show that the initial IKE setup and periodic key-changes in IKEv2 SA spends about the same time changing keys as with SD-SA (in average 3.6 ms vs 4.0 ms in Table: 10.2). However, IKE SA demands more computation resources than SD-SA, such as 31,0% vs 2,5% CPU time Table: 10.3)

A consequence of this is also seen when running IKE reconnections with a 2-second interval, where the whole process congests and measuring is not possible. This is because the process is not finished before a new key-change thread is initiated (Table: 10.2).

IKE child SA key-change

Traditional IKEv2 with child SA key-changes (Phase 2 in IKEv1) has less time period of packet loss than non-IKE with SD-SA (1.6ms versus 4.0ms in Table: 10.2). However, this difference is not reflected in the TCP bandwidth test, where SD-SA, in fact, performs better than IKEv2 (682 Mbps vs 702 Mbps in Table: 10.2).

A child SA key-change is more frequent than an SA key-change, but for SD-SA

there is no difference in these type key-changes. Hence, the performance results of child SAs are considered more important and relevant. However, in virtualised environments with dynamic resource allocation, it is also reasonable to have many IKE reconnections due to Virtual Machine migrations [66].

Throughput

Interestingly, the performance tests show that the throughput is not significantly affected by the number of key-changes (Figure: 10.2). By monitoring the process consumption, we discovered that our Ubuntu operating system automatically dedicated CPU resources to data plane packet-handling (processor one) and control plane key-changes (processor two) respectively. The Linux kernel on the VPN peers spend most of the CPU time on handling software interrupts when running VPN data traffic through them. This causes one of the CPU to peak close to a 100% utilisation while performing packet encryption. The reduction of 168 bytes of available packet size is another reason for the system not to be able to archive 717 Mbps with no key-change.

Resource consumption

With respect to resource consumption, we measured memory and CPU time without running any data plane traffic. Table: 10.3 shows that IKEv2 reconnections require a significant amount of CPU time. Both 30 and 1000 connections were not able to finish within the time intervals of 4 and 30 seconds. Therefore these measurements are not precise with respect to memory consumption per connection. The main reason for this delay is the waiting time for network packets from the remote peer. Hence, this result indicates that SD SA is a much more efficient way to reconnect IPsec sessions.

The reason behind the significant differences in memory consumption across the tests is that the IPsec application Strongswan prepares and installs the next keys in advance. Therefore the memory consumption is doubled. In our simplified experiment, we did not consider SA overlapping during key-changes for SD-SA. However, it is possible to have two incoming ESP packet policies that ensure that the VPN peer receiver does not delete the old IPsec policy before a new one is active. We chose not to implement this feature for both IPsec/SD-IKE and IPsec/SD-SA. In respect to the measurements in the experiment, this parallelism feature would not enable a precise measurement of the time gap for key-change delays. However, it is assumed that overlapping IPSec policies would reduce the packet loss. The main difference between the IPsec based on IKE or SD-SA would concern the amount of resource consumption.

Latency: Another factor that influences the performance is the distance and latency

between the VNFs. In the presented test scenario the virtual machines are provisioned in a closed environment with direct peerings between the hosts, that enables a very low latency between the hosts. If the SDN controller is placed outside of the data centre, or if the VPN peers exist in different data centres, this would increase the latency between the peers and consequently affect the delay between the key-changes on each VPN peer. However, It is expected that this delay is similar for IPsec with IKE and for SD-SA.

This is because the SDN controller sends a similar key update to both the SD-SA application and the IKE application and therefore the difference should be equally linearly to the latency. Also, for this proof of concept demonstration, we primarily aim to show that SD SA works equally or better than IKE in order to further apply it to an NFV domain. For that reason, this latency is not highly relevant when comparing the methods.

Communication constraints

Concerning performance, IKE and SD-SA perform relatively similar, but SD-SA performs slightly better when the number of key-changes is high. We have shown that both IPsec enabled with IKE and with SD-SA can ensure isolation of Virtual Links in federated NFV environments. However, SD-SA consumes less resources and has the advantage of not being dependent on the transport protocol on the data plane. For example, if the data plane transport channel is based on NSH, direct communication between the VNFs is not possible over NSH. Hence, a separate control-channel is required in order to make the VPN peers (the VNFs) exchange keys. This implies that, regardless of performance, IPsec with SD-SA is more suitable in federated NFV domains due to these communication constraints. However, for comparison reasons, we did not use NSH on the data plane in our performance test

Benefits of SDN

Our approach of separating IKE from packet encryption is similar to related results performed by Vajaranta et al. [67]. Their experiment was based on utilising OpenFlow to load-balance IPsec. Their results showed that, for high bandwidths in particular, OpenFlow enhances the IPsec availability and performance. Our experiment was based on NFV with a secure distribution of keys and not based on OpenFlow as a load-balancer. However, the distributed design and flow-based control of SDN for both experiments emphasise the scalability benefit in distributed environments. This confirms our results, but it also indicates that our design of secure key distribution is applicable to other application domains such as load-balancing of IPsec.

IKE drawbacks in Virtualised environments

The consequence of VM migration and dynamic resource allocation in virtual environments also favorises SD-SA in front of IKE. Having encryption services running as VNFs implies that the encryption services can migrate between hosts and change virtual machines along with SFC changes. Every VM migration would require an IKE reconnection, while SD-SA only requires a key update if we follow our suggested architecture [1]. This clearly distinguishes VPN setups in virtual infrastructure domains and hardware-based VPN networks. If an SDN controller is responsible for both distributing encryption keys and performing routing, it is expected that such topologies can reduce the failover time compared to traditional BGP routing [68] and IKE IPSec. We aim to test this in our future work (see Section: 10.6.4).

10.6.2 Security analysis

In this Section, we analyse the security of our proposal. Due to the use of multiple protocols, components and communication planes, a formal method or a code analysis is difficult to archive in order to analyse the level of security. Also, the operational characteristics of the key exchange mechanism are not fully specified, that makes a formal verification difficult. However, we did analyse standard network protocol security features and resistance against well-known attacks according to basic security principles [69] such as confidentiality, integrity, availability.

Confidentiality

The objective of our security mechanism is to keep the encryption key and the integrity key for the SA in IPsec private. For IKEv2, the peers derive these keys between each other from parameters such as the preshared key, while in SD-SA the keys are sent directly to the peers from the controller. Both methods require that the configuration channel between the controller and the VPN peer is protected. In both scenarios, the configuration channel is protected by SSL. This means that the underlying keys are dependent on the integrity and confidentiality of the configuration channel. This implies both the protection of the network, such as the quality of the ciphering algorithm, but also the protection of software components. The system fully relies on the orchestrator in distributing the keys to the VNFs. A compromised VNF or a compromised orchestrator therefore breaks the security and enables the adversary to launch attacks using the keys obtained.

Both IKE v1 and IKE v2 focus on multiple iterations of key derivations to make sure that the encryption key and the integrity keys are kept confidential. The keys are not transferred between the peers such as we suggested for SD-SA. However, for future extensions, the SD-SA key transfers are also possible to extend with ad-

ditional key derivations such as Diffie-Hellman [70]. In addition to key derivations in IKE, the keys are also kept protected inside the kernel and the IPsec application and not shown in a configuration file. This makes the encryption key and the integrity key less available in a system that is not fully compromised.

We have not investigated how the security mechanism can be protected from a compromised SDN controller, authentication centre, orchestration system or VNF. However, it is assumed that additional security features such as integrity attestation of software packages must co-exist with the presented key exchange mechanism.

Integrity

It is not investigated how replay attacks are possible over the configurations channels. However, the NETconf protocol states that the underlying transport protocol must handle such protection [71].

Scalability

The number of controllers and the number of virtual machines can easily be adjusted in a virtualised environment, However, the computational resources needed for encryption and decryption is closely connected to how much data traffic the end-users are consuming. Sudden changes in behavioural patterns, such as viral videos, could potentially demand more computational power than available in the NFV domain. Virtual environments use shared resources and often overbooked services. In use cases where each user runs SFCs with multiple encrypted Virtual Links, the computational need for performing encryption is exponential to the bandwidth utilisation and number of Virtual Links.

Availability

Both the VPN peers and the SDN controller is vulnerable for Denial of Service attacks. Especially, DDoS towards VPN peers can result in amplified resource consumption as mentioned in the previous section. For DDoS towards the SDN controller, we assume that it runs in a federated control plane domain [1] separated for the data plane. Hence, the attack surface is considered relatively low. However, a DDoS attack on the data plane in multiple VPN channels will increase the CPU resource consumption for all types of encrypted topologies. Hence, this problem will affect both IPsec/IKE, IPsec/SD-SA or other underlying IPsec channels in an SFC with a similar amount of resource consumption.

Another aspect of availability is network attacks on the IKE UDP port 500. Half-open IKE connection is a resource consumption problem that comes from establishing too many IKE connection. Because the IKE protocol often runs over a network port available on the data plane, IKE is more vulnerable for such attacks

than SD-SA. This is because SD-SA is suggested to run over a control plane network that is separated from the data plane.

Reliability

Our implementation did not take into account the aforementioned scenarios where the SDN controller or other components become unavailable. For example, if the controller becomes unavailable, it is important that the VPN peers do not use the cipher key after the lifetime expires. For such cases, the local configuration service in the VPN peer has to ensure that the key expires if no key is received from the SDN controller. We did neither consider system responses to deadlock or system crash in any of the components.

10.6.3 Interoperability

This article suggests a new key distribution paradigm for the encryption of Virtual Links per SFC in NFV. This raises an interoperability problem of both the VNFs and the NFV infrastructure components. From a NFV infrastructure perspective, the suggested architecture is proposed in interconnected and federated NFV domains. This implies that the underlying infrastructures support multi-tenant domains, where each tenant, in theory, is capable of running their own customer-specific SFC routing method, supported by a network overlay. However, VNF interoperability is more challenging. According to the SFC specification, the VNFs can be (1) SFC-aware or (2) SFC-unaware supported by an SFC aware proxy. This implies that the SFC proxy or the VNF must be aware of the SFC routing mechanism, such as the NSH headers. Consequently, the encrypting VNFs must also support SFC routing. The different SFC routing methods, such as segment routing in MPLS, IPv6 or NSH, put a burden on VNF developers in supporting different standards. This is a general VNF problem and not specific to our application. However, our application introduces an additional parameter for the VNF developer to consider. It also raises a new standardisation issue of encryption applications and their programming interface towards the AuC and how to deal with different types the SFC data forwarding standards. Hence, this proof of concept experiment aims to contribute to the standardisation of VNF application interfaces for enabling encrypted Virtual Links. This also indicates a need for a standardised encryption header in the SFC protocol, such as NSH.

10.6.4 Future work

In this article, we chose to focus on the security mechanism of the key-exchange between virtual encryption functions for providing isolated SFCs. We did not take SFC transport mechanisms such as MPLS or NSH into account when we performed our measurements. Neither did we consider topology changes and the

effect of routing protocols delays in our design. For future work, the routing protocol of the SFCs needs to be aware of the cryptographic endpoints for every hop in an SFC. This brings an additional cryptographic attribute the NFV routing and resource allocation problem [72]. This optimization problem is an NP-hard problem that we aim to resolve by distributing the routing decisions by the use of multi-protocol BGP [68]. Consequently, our future work relies on providing a testbed for integrating the encryption functions and the routing mechanism into an NFV testbed.

10.7 Conclusion

We have presented a new way of utilising SDN in NFV by automating the key distribution in the setup of secure VPN channels between VNFs. This mechanism was specifically developed in order to enable per-flow encryption in federated NFV environments. However, it also solves the communication problem of establishing an IKE Security Association between two VNFs in an SFC. Through our proof of concept demonstration, we have shown that the automation procedure can be utilised to setup Security Associations between VPN peers for both IKE and non-IKE IPsec VPN connections. However, in comparison, the proposed SD-SA mechanism can be even more efficient and scalable than traditional IKE. The results of the performance tests show that the delay between rekeying the VPN peers are slightly faster when running IKE, while SD-SA requires less resources. The presented architecture can be utilised both for small and large data centre deployments. However, the automated key distribution mechanism is excepted to have the greatest benefit in an NFV environment, with a lot of encrypted channels, such as in a per-flow per service chain encryption. The proposed bootstrapped mechanism is based on standard internet security protocols such as HTTPS and RESTconf where the majority of the security relies on these underlying protocols. We have seen that the most critical factor for our proposal is to have available compute resources for encryption and decryption.

Funding:

This research was funded by Eidsiva, the Norwegian Research Council (Project number: 25127) and the Norges Teknisk- Naturvitenskapelige Universitet .

Conflicts of Interest:

The authors declare no conflict of interest.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi: 10.1016/j.comnet.2019.05.015 .

CRediT authorship contribution statement

Håkon Gunleifsen: Conceptualization, Writing - original draft.
Thomas Kemmerich: Writing - review & editing, Methodology.
Vasileios Gkioulos: Writing - review & editing, Methodology.

References

[1] Håkon Gunleifsen, Vasileios Gkioulos and Thomas Kemmerich. 'A Tiered Control Plane Model for Service Function Chaining Isolation'. In: *Future Internet* 10.6 (2018), p. 46.

[2] Håkon Gunleifsen and Thomas Kemmerich. 'Security requirements for service function chaining isolation and encryption'. In: *IEEE 17th International Conference on Communication Technology (ICCT)*. 2017, pp. 1360–1365.

[3] Håkon Gunleifsen, Thomas Kemmerich and Slobodan Petrovic. 'An End-to-End Security Model of Inter-Domain Communication in Network Function Virtualization'. In: *Norsk Informasjonssikkerhetskonferanse (NISK): Bergen, Norway* (2016), pp. 7–18.

[4] ETSI. *Network Function Virtualization (NFV) Architectural Framework v1.1.1*. Available online: http://www.etsi.org/deliver/etsi_gs/NFV/001_ 099 /002/01.01.01_60/gs_NFV002v010101p.pdf (accessed on 23 August 2019). 2014.

[5] Norival Figueira et al. 'Policy Architecture and Framework for NFV Infrastructures'. In: *Active Internet-Draft, IETF Secretariat, Internet-Draft draft-irtf-nfvrg-nfv-policy-arch-01* (2015).

[6] Joel M. Halpern and Carlos Pignataro. *Service Function Chaining (SFC) Architecture*. RFC 7665. Oct. 2015.

[7] Vishwas Manral and Stephen R. Hanna. *Auto-Discovery VPN Problem Statement and Requirements*. RFC 7018. Sept. 2013.

[8] Sangwon Hyun et al. *I2NSF Registration Interface Data Model*. Internet-Draft draft-hyun-i2nsf-registration-interface-dm-06. Work in Progress. Internet Engineering Task Force, July 2018. 23 pp.

[9] Diego Lopez et al. *Framework for Interface to Network Security Functions*. RFC 8329. Feb. 2018.

[10] Alia Atlas et al. *An Architecture for the Interface to the Routing System*. RFC 7921. June 2016.

[11] Paul Quinn, Uri Elzur and Carlos Pignataro. *Network Service Header (NSH)*. RFC 8300. Jan. 2018.

[12] Ahmed M Alwakeel, Abdulrahman K Alnaim and Eduardo B Fernandez. 'A Survey of Network Function Virtualization Security'. In: *SoutheastCon 2018, St. Petersburg, FL, USA*. IEEE. 2018, pp. 1–8.

[13] Ibrahim Afolabi et al. 'Network slicing and softwarization: A survey on principles, enabling technologies, and solutions'. In: *IEEE Communications Surveys & Tutorials* 20.3 (2018), pp. 2429–2453.

[14] Mahdi Daghmehchi Firoozjaei et al. 'Security challenges with network functions virtualization'. In: *Future Generation Computer Systems* 67 (2017), pp. 315–324.

[15] Shankar Lal, Tarik Taleb and Ashutosh Dutta. 'NFV: Security threats and best practices'. In: *IEEE Communications Magazine* 55.8 (2017), pp. 211–217.

[16] David Naylor et al. 'Multi-context TLS (mcTLS): Enabling secure in-network functionality in TLS'. In: *ACM SIGCOMM Computer Communication Review* 45.4 (2015), pp. 199–212.

[17] Karthikeyan Bhargavan et al. 'A formal treatment of accountable proxying over TLS'. In: *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2018, pp. 799–816.

[18] Jari Arkko, Vesa Lehtovirta and Pasi Eronen. *Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)*. RFC 5448. May 2009.

[19] Dr. Clifford Neuman et al. *The Kerberos Network Authentication Service (V5)*. RFC 4120. July 2005.

[20] Rafael Lopez and Gabriel Lopez-Millan. *Software-Defined Networking (SDN)-based IPsec Flow Protection*. Internet-Draft draft-ietf-i2nsf-sdn-ipsec-flow-protection-02. Work in Progress. Internet Engineering Task Force, July 2018. 48 pp.

[21] Karen Seo and Stephen Kent. *Security Architecture for the Internet Protocol*. RFC 4301. Dec. 2005.

[22] Paul E. Hoffman. *Algorithms for Internet Key Exchange version 1 (IKEv1)*. RFC 4109. May 2005.

[23] Pasi Eronen et al. *Internet Key Exchange Protocol Version 2 (IKEv2)*. RFC 5996. Sept. 2010.

[24] Tirumaleswar Reddy et al. *Authenticated and encrypted NSH service chains*. Internet-Draft draft-reddy-sfc-nsh-encrypt-00. Work in Progress. Internet Engineering Task Force, Apr. 2015. 12 pp.

[25] I Kotuliak, P Rybár and P Trúchly. 'Performance comparison of IPsec and TLS based VPN technologies'. In: *9th International Conference on Emerging eLearning Technologies and Applications (ICETA)*. IEEE. 2011, pp. 217–221.

[26] Jason A Donenfeld. 'WireGuard: next generation kernel network tunnel'. In: *24th Annual Network and Distributed System Security Symposium, NDSS*. 2017.

[27] Anna Selvåg Braadland. 'Key Management for Data Plane Encryption in SDN Using WireGuard'. MA thesis. NTNU, Norway, 2017.

[28] Sangwon Hyun et al. 'Interface to Network Security Functions for Cloud-Based Security Services'. In: *IEEE Communications Magazine* 56.1 (2018), pp. 171–178.

[29] Kristian Malmkvist Eie. 'Authentication in Protected Core Networking'. MA thesis. NTNU, Norway, 2016.

[30] Nobuo Okabe et al. 'Implementing a secure autonomous bootstrap mechanism for control networks'. In: *IEICE Transactions on Information and Systems* 89.12 (2006), pp. 2822–2830.

[31] Juhani Latvakoski et al. 'A survey on m2m service networks'. In: *Computers* 3.4 (2014), pp. 130–173.

[32] Lidong Zhou and Zygmunt J Haas. 'Securing ad hoc networks'. In: *IEEE network* 13.6 (1999), pp. 24–30.

[33] Nidal Aboudagga et al. 'Authentication protocols for ad hoc networks: taxonomy and research issues'. In: *Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*. ACM. 2005, pp. 96–104.

[34] Yiran Gao, Chris Phillips and Liwen He. 'DVM Based Dynamic VPN Architecture for Group Working and Orchestrated Distributed Computing'. In: *Third International Conference on Digital Information Management, 2008. ICDIM 2008.* IEEE. 2008, pp. 763–768.

[35] Shardul Gokhale and Partha Dasgupta. 'Distributed authentication for peer-to-peer networks'. In: *2003 Symposium on Applications and the Internet Workshops*. IEEE. 2003, pp. 347–353.

[36] Sadanori Aoyagi et al. 'ELA: a fully distributed VPN system over peer-to-peer network'. In: *The 2005 Symposium on Applications and the Internet*. IEEE. 2005, pp. 89–92.

[37] Sencun Zhu, Sanjeev Setia and Sushil Jajodia. 'LEAP+: Efficient security mechanisms for large-scale distributed sensor networks'. In: *ACM Transactions on Sensor Networks* 2.4 (2006), pp. 500–528.

[38] William A. Simpson. *PPP Authentication Protocols*. RFC 1334. Oct. 1992.

[39] William A. Simpson. *PPP Challenge Handshake Authentication Protocol (CHAP)*. RFC 1994. Aug. 1996.

[40] Glen Zorn. *Microsoft PPP CHAP Extensions, Version 2*. RFC 2759. Jan. 2000.

[41] Allan Rubens et al. *Remote Authentication Dial In User Service (RADIUS)*. RFC 2865. June 2000.

[42] Thorsten Dahm et al. *The TACACS+ Protocol*. Internet-Draft draft-ietf-opsawg-tacacs-11. Work in Progress. Internet Engineering Task Force, Sept. 2018. 44 pp.

[43] Pat R. Calhoun et al. *Diameter Base Protocol*. RFC 3588. Sept. 2003.

[44] Daniel Simon, Bernard D. Aboba and Pasi Eronen. *Extensible Authentication Protocol (EAP) Key Management Framework*. RFC 5247. Aug. 2008.

[45] Ashwin Palekar et al. *Protected EAP Protocol (PEAP) Version 2*. Internet-Draft draft-josefsson-pppext-eap-tls-eap-10. Work in Progress. Internet Engineering Task Force, Oct. 2004. 87 pp.

[46] Eric Rescorla and Tim Dierks. *The Transport Layer Security (TLS) Protocol Version 1.2*. RFC 5246. Aug. 2008.

[47] Eric Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. Aug. 2018.

[48] Eric Rescorla and Nagendra Modadugu. *Datagram Transport Layer Security Version 1.2*. RFC 6347. Jan. 2012.

[49] Brian Campbell, Chuck Mortimore and Michael Jones. *Security Assertion Markup Language (SAML) 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants*. RFC 7522. May 2015.

[50] Dick Hardt. *The OAuth 2.0 Authorization Framework*. RFC 6749. Oct. 2012.

[51] Frederic Detienne, Manish Kumar and Mike Sullenberger. *Flexible Dynamic Mesh VPN*. Internet-Draft draft-detienne-dmvpn-01. Work in Progress. Internet Engineering Task Force, Dec. 2013. 32 pp.

[52] Daniel Simon, Dr. Bernard D. Aboba Ph.D. and Pasi Eronen. *Extensible Authentication Protocol (EAP) Key Management Framework*. RFC 5247. Aug. 2008.

[53] Hugh Harney et al. *GSAKMP: Group Secure Association Key Management Protocol*. RFC 4535. June 2006.

[54] John G. Myers. *Simple Authentication and Security Layer (SASL)*. RFC 2222. Jan. 1997.

[55] Derrell Piper and Brian Swander. *A GSS-API Authentication Method for IKE*. Internet-Draft draft-ietf-ipsec-isakmp-gss-auth-07. Work in Progress. Internet Engineering Task Force, July 2001. 13 pp.

[56] J Vilhuber et al. *Kerberized Internet Negotiation of Keys (KINK)*. RFC 4430. Mar. 2006.

[57] Mohamed Khalid et al. *VPN processing via service insertion architecture*. US Patent 8,429,400. Apr. 2013.

[58] Donald McAlister and John Cary Orange. *Protocol/API between a key server (KAP) and an enforcement point (PEP)*. US Patent App. 11/541,424. Apr. 2008.

[59] Michael L Sullenberger and Jan Vilhuber. *Method and apparatus for establishing a Dynamic Multipoint encrypted Virtual Private Network*. US Patent 7,447,901. Nov. 2008.

[60] Yoav Nir and Qin Wu. *An Internet Key Exchange Protocol Version 2 (IKEv2) Extension to Support EAP Re-authentication Protocol (ERP)*. RFC 6867. Jan. 2013.

[61] Fushan Wei, Chuangui Ma and Zhenfeng Zhang. 'Gateway-oriented password-authenticated key exchange protocol with stronger security'. In: *International Conference on Provable Security*. Springer. 2011, pp. 366–379.

[62] Ken'ichi Kamada et al. *Problem Statement on the Cross-Realm Operation of Kerberos*. RFC 5868. May 2010.

[63] Andy Bierman, Martin Björklund and Kent Watsen. *RESTConf Protocol*. RFC 8040. Jan. 2017.

[64] Honglei Wang and Xia Chen. *Yang Data Model for Internet Protocol Security (IPsec)*. Internet-Draft draft-tran-ipsecme-yang-01. Work in Progress. Internet Engineering Task Force, Mar. 2016. 99 pp.

[65] H Soussi et al. 'IKEv1 and IKEv2: A quantitative analyses'. In: *Proceedings of World Academy of Science, Engineering and Technology*. Vol. 6. 2005, pp. 194–197.

[66] Christopher Clark et al. 'Live migration of virtual machines'. In: *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation-Volume 2*. USENIX Association. 2005, pp. 273–286.

[67] Markku Vajaranta, Joona Kannisto and Jarmo Harju. 'IPsec and IKE as Functions in SDN Controlled Network'. In: *Network and System Security*. Springer International Publishing, 2017, pp. 521–530.

[68] Yakov Rekhter and Tony Li. *A Border Gateway Protocol 4 (BGP-4)*. RFC 1654. July 1994.

[69] Carl A Sunshine. 'Survey of protocol definition and verification techniques'. In: *ACM SIGCOMM Computer Communication Review* 8.3 (1978), pp. 35–41.

[70] Eric Rescorla. *Diffie-Hellman Key Agreement Method*. RFC 2631. June 1999.

[71] Rob Enns et al. *Network Configuration Protocol (NETConf)*. RFC 6241. June 2011.

[72] Hao Feng et al. 'Approximation algorithms for the NFV service distribution problem'. In: *Conference on Computer Communications (Infocom)*. IEEE. 2017, pp. 1–9.

Chapter 11

A Proof-of-Concept Demonstration of Isolated and Encrypted Service Function Chains

Published in Multidisciplinary Digital Publishing Institute (MDPI) Journal; Future Internet, 2019

Håkon Gunleifsen, Thomas Kemmerich and Vasileios Gkioulos

Department of Information Security and Communication Technology, Norwegian University of Science and Technology (NTNU), Postbox 191, 2802 Gjøvik, Norway

hakon.gunleifsen2@ntnu.no, thomas.kemmerich@ntnu.no, vasileios.gkioulos@ntnu.no

Abstract

Contemporary Service Function Chaining (SFC), and the requirements arising from privacy concerns, call for the increasing integration of security features such as encryption and isolation across Network Function Virtualisation (NFV) domains. Therefore, suitable adaptations of automation and encryption concepts for the development of interconnected data centre infrastructures are essential. Nevertheless, packet isolation constraints related to the current NFV infrastructure and SFC protocols, render current NFV standards insecure. Accordingly, the goal of our work was an experimental demonstration of a new SFC packet forwarding standard that enables contemporary data centres to overcome these constraints. This article presents a comprehensive view of the developed architecture, focusing on

the elements that constitute a new forwarding standard of encrypted SFC packets. Through a Proof-of-Concept demonstration, we present our closing experimental results of how the architecture fulfils the requirements defined in our use case.

Keywords: NFV; SFC; NSH; IPsec; P4; RESTconf

11.1 Introduction

The current Service Function Chaining (SFC) architecture suggested by the European Telecommunications Standards Institute (ETSI) [1] lacks the capability to encrypt and isolate end-user traffic between Service Functions (SFs) in Network Function Virtualisation (NFV). End-to-end encryption of end-user traffic is by design impossible when middleboxes such as SFs require access to the data content of the packets. This constraint in NFV questions how confidentiality can be integrated into an SFC. The scope of our work is to cover this gap, by enabling automated hop-by-hop encryption in an SFC. We aim for contemporary data centre networks to support an architecture of nested SFC tunnels in order to support hop-by-hop encryption within the current NFV [1] and SFC [2] standards. As presented in our previous work [3, 4], the current packet forwarding standards do not support SFC forwarding of encrypted packets because the relevant packet headers for SFC routing are also encrypted. Accordingly, our work explicitly focused on these constraints, aiming initially to provide a Proof-of-Concept for the capacity to deploy a secure architecture as an overlay to the existing NFV infrastructures.

Under this scope, our security-related studies followed five consecutive steps (Figure 11.1), following the Design Science Research Methodology (DSRM) defined by Peffers et al. [5]. Initially (A), the operational constraints and the NFV forwarding standards were surveyed [6]. Consequently (B), the security requirements have been identified aiming to accommodate the requirements extracted from the aforementioned studies [4]. Thirdly (C), an automated forwarding architecture has been developed based on a web service architecture, aiming to accommodate the requirements and the constraints [3]. The fourth step (D) in our studies was to develop a security protocol for exchanging encryption keys between SFs [7]. This article (E) integrates the previous results into a customised NFV environment and combines it with SFC routing [8]. In order to overcome the network constraints, we developed a customised virtual switch by the use of P4 [9] in order to support a new SFC packet header based on Network Service Headers (NSH). Accordingly, we aim to verify that this implementation fulfils the requirements we have developed in our previous work.

Section 11.2 summarises the related work to this research. Section 11.3 presents

the operational context under which the developed architecture was designed. Sections 11.4 and 11.5 present the architecture and implementation, while Section 11.6 gives a verification of the presented scenarios for a closing demonstration. Through defining three episodes in this scenario, we seek to highlight how the elements presented in this paper are supporting a secure SFC implementation in NFV.

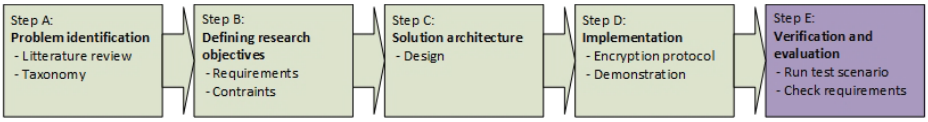


Figure 11.1: Research method.

11.2 Related Work

In recent years, NFV based solutions have become a very active research area, due to the benefits promised by cost-effective solutions when virtualising network equipment. Within the research area of NFV, SFC forwarding protocols and their corresponding control plane mechanisms are NFV research areas that have gained attention [10, 11, 12]. Nevertheless, the security research on these networking standards is limited, where none of the SFC standards are protecting the privacy and the integrity the data plane traffic [13]. This is a complex problem that consists of data protection problems on multiple levels; the orchestration plane, the control plane and the data plane. Hence, our work aims to cover this research gap, by providing a new packet forwarding standard that is reflected on all these planes.

From a data plane perspective, the Internet Engineering Task Force (IETF) workgroup NVO3 [14] have considered multiple overlay protocol for use in data centres. Generic UDP Encapsulation [15], Geneve [14], VXLAN-GPE [16], and NSH [17] are all protocol competing to be the next standard. They all have limitations related to multi-vendor and multi-domain interoperability and they also have a lack of security extensions. MPLS-SR [18] does support multi-domain topologies, but, in an SFC context, they all rely on the underlying protocol, such as IPSec tunnels, to provide encryption. However, such a tunnel can be perceived as a wire between data centres. Multiples of these tunnels constitute a virtual overlay network that is unprotected from all data threats that reside within the network. Then, there is no protection of the integrity of the headers of the data-flow across multiple domains. In IPv6, Segment Routing (SR) [8] is supported; however, during encryption, the header is replaced and the segments become invisible for intermediate routers. We aim to solve this by introducing a new overlay packet header that supports encryption inside the overlying network protocols, such as NSH.

With respect to interconnected control planes, we have earlier showed [6] that there are two orchestration methods across multiple service provider domains. (1) A top-down approach by utilising a hierarchy of orchestration planes [19] or control planes [20] or (2) by using an east–west control plane approach such as SDNi [21] or BGP [22]. Both interconnection methods try to overcome the problem of multiple forwarding standards and multiple types of network controllers. The industry has responded by providing tenant-based data centres, where each tenant extends their data centre across multiple sites and omits the need for control plane interconnections. Networking by NSX-T [23] from VMWare is one example of such multi-tenant data centre technologies where a micro-segmented infrastructure can span over multiple sites. However, most of the underlying network protocols, such as Geneve [14] in NSX-T, are not capable of combining SR with micro-segmentation and flow-based encryption. Hence, we have in our previous work [3] suggested a new SFC header, based on an NSH extension that adds more granularity to the security aspect of an SFC. Correspondingly, we have in this paper developed a RESTconf based control plane for distributing the forwarding decisions of this new packet header.

Introducing a new packet header has historically been problematic with respect to the adoption into existing hardware. When a new network protocol was suggested, the network operators had to wait for a set of standardisation documents from organisations such as IETF and ONF [24]. Furthermore, they also had to wait for the switch vendors to develop a new software version. Sometimes, a new network standard also required new hardware. The Programming Protocol-independent Packet Processors (P4) [25] language aims to solve this issue by defining a framework that directly programs packet parsing and packet forwarding instructions to a switch in runtime. Then, network operators themselves can program their switches and add new protocols and features to them. The ONF group is currently aiming for standardising P4 as a part of SDN through the Stratum project [24]. In this research, we run P4 inside a Virtual Machine in order to simulate a virtual switch. Due to the lack of OpenFlow implementations in our P4 framework, we used RESTconf for the control plane protocol.

From the encryption perspective, no protocols have been found for providing micro-segmented and flow-based encryption per SFC. However, our previous work [7] that originated from Software-Defined IPsec Flow Protection in SDN [26] and IPsec Key Exchange using a Controller [27], showed how encryption and Software-Defined Security Associations (SD-SA) could be adapted to an NFV domain.

In this paper, we combine this SD-SA encryption architecture [7] with our new SFC header [3] and a new flow distribution control plane. The security features of the architecture are verified by demonstrating how the requirements such as

isolation and encryption comply with a use case scenario.

11.3 Operational Context of Proof-of-Concept Scenarios

This section presents a use case scenario of SFC isolation and encryption. Furthermore, we show three episodes of this scenario that are developed based on a set of architectural security requirements.

11.3.1 Use Case

The verification scenario for our Proof-of-Concept demonstration is based on a fictional Internet Service Provider (ISP) that wants to extend their NFV portfolio and their data centre resources. The ISP located in country A, named ISP-A, wants to lower their costs on their Customer Premise Equipment (CPEs) by virtualising them and consequently more efficiently extending their service delivery. They have limited resources in their data centre and want to offload parts of their services to remote data centres. They have found two cooperative partners in country B (ISP-B) and country C (ISP-C) that can provide them with data centre resources. They want all data centres to contribute to delivering and extending their virtual CPE (vCPE) services. They are aiming to provide this by chaining SFs across all data centres by the use of the SFC protocol NSH.

The IETF has defined a variety of SFC use cases [28], but, for our Proof-of-Concept demonstration, we limit the SFC use case to the following: ISP-A aims to provide three SFs to their customers. Two of the SFs are mandatory, while one additional SF is optional for the end-users to choose. The basic SFs are a vCPE (SF-1) and a firewall (SF-3), while the optional SF is a video caching service (SF-2). Due to the cost of data centre resource consumption and SF security policies, the ISP-A policy is defined to require that the vCPE runs at ISP-A, the video caching service at ISP-B and the firewall at ISP-C. The vCPE is the first element in the SFC. The additional video caching service is placed in the middle of the SFC in order to let the first two services be protected by the last element in the SFC, which is a virtual firewall (SF-3) (Figure 11.2). Hence, from a service plane perspective, the firewall is protecting the inner SFs and the end-user from the outside world.

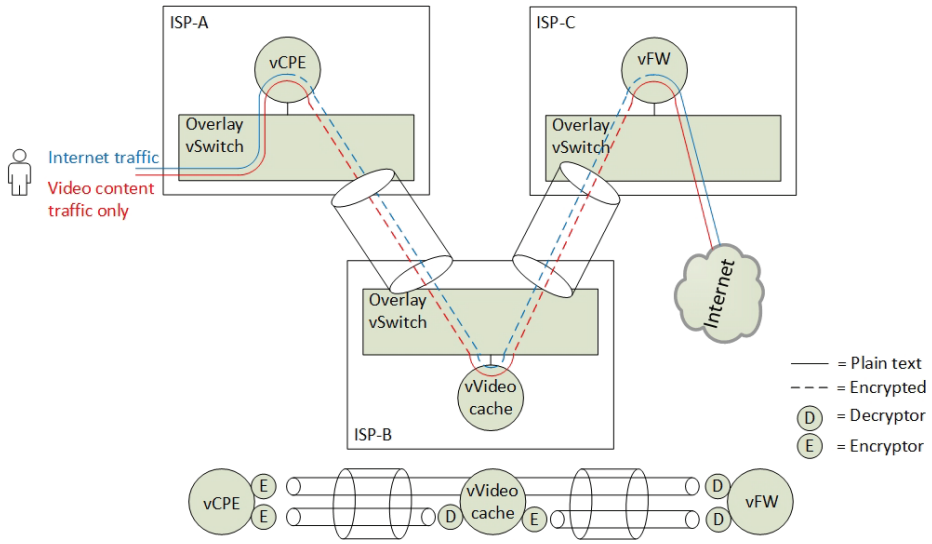


Figure 11.2: Proof-of-Concept scenario.

The data centres operate with multi-tenants where ISP-A has interconnected all their tenant instances in an overlay network. In this setup, ISP-A is concerned about the privacy of their customers and they do not know if ISP-B is eavesdropping the end-user traffic traversing them. Neither are they sure whether ISP-B is malicious. ISP-C is, on the other hand, a trusted partner. In order not to let ISP-B being capable of eavesdropping all end-user traffic, all traffic that is traversing ISP-B, except video streaming traffic, must be encrypted. Note: for Proof-of-Concept purposes, the video caching service is categorised as a non-privacy sensitive service.

We simplified the SFC isolation problem by omitting a full mesh topology of the interconnected data centres in the Proof-of-Concept scenario. However, the components in a full mesh topology are also vulnerable to eavesdropping. Corrupt intermediate virtual switches or faulty SFs can modify, intercept and manipulate SFC traffic inside an overlay network. Hence, the protection of the Virtual Link (VL) [2] is relevant both between the Compute Nodes in one data centre and for the VLs between multiple data centres.

11.3.2 Requirements

We have in our previous work presented the NFV security requirements [4] for the encryption and the isolation of the VLs. We summarise these requirements in the context of the aforementioned scenario:

i. **Hop by hop encryption**—In order to prevent eavesdropping of the VLs, the VLs must be encrypted per SFC.

ii. **Micro-segmented isolation**—The SFC specification [2] does not allow micro-segmentation within one SFC. However, we state that the end-user requires that they must be able to specify what data traffic the SFs are allowed to handle on a flow-based level. Hence, it is required that the associated data plane components are capable of isolating different packet flows with different encryption keys within one single SFC.

iii. **Header visibility**—When encrypting VLs, the SFC packet header must be non-encrypted in order to enable SFC routing of the encrypted packets. We define that a new SFC header extension must be able to both allow and specify when the inner data-content of an SFC packet header is encrypted.

iv. **Control plane flow distribution**—Multiple encryption-flows within one SFC require that the control plane is capable of distributing route information about each of these encrypted flows. These flow-rules must be securely distributed. Hence, secure and trusted intra- and inter-domain communication channels from the virtual network devices to the network controllers must be established.

v. **Key distribution**—Due to a non-bidirectional data plane between the SFs in an SFC [2], a new hop-by-hop key distribution mechanism is required. The key distribution mechanism must respond to a dynamic SFC behaviour such as an SFC modification. It must also support future encryption standards or protocol extensions. The key distribution mechanism must ensure confidentiality, integrity and availability of the keys.

vi. **Compliance and adoption**—A new SFC header and the corresponding provisioning architecture must be compliant with the current NFV standards. In addition, adding encryption to the VLs should not degrade the end-to-end throughput performance more than traditional end-to-end encrypted channels. Another important factor for the architecture to be adopted is that the end-users do not perceive a significant increase in service provisioning times when they apply VL encryption.

vii. **Resilience and availability**—The architecture must provide resilience towards components failing without reducing the level of security.

viii. **Security integrity**—An attacker should not be able to manipulate the routing tables or to modify the packet headers in order to enforce access to non-encrypted data packets.

We aggregated these requirements and defined the following episodes from the aforementioned scenario.

Episode 1: Packet forwarding and provisioning (Req: i, iii, iv, vi)

This episode is created from an end-user perspective. An end-user orders a new virtual service according to our scenario. The end-user expects that his broadband service is not affected during service provisioning. A service provisioning demonstration can monitor the provisioning time by measuring network outage. However, demonstrating a full service provisioning also provides evidence of how the architecture provides the setup of the encrypted VLs. In addition, in a fully provisioned SFC, an end-to-end traffic test shows how the encrypted data packets are routed and if the traffic flow is satisfying the security requirement of flow-distribution.

Episode 2: Resilience and availability (Req: v, vii)

In this episode, we simulate hardware failure. From an availability and resilience perspective, the architecture must be resilient to components failing without compromising the network encryption policy. During service recovery, this demonstration also shows the dynamic behaviour of the key distribution during failovers.

Episode 3: Security integrity (Req: ii, viii)

For our third episode, we simulate that one of our data centres (ISP-B) is attacked and that a subset of the components is compromised. When simulating a set of basic network attacks, the architecture must be resistant to this. This also includes a demonstration of how flow-based encryption can protect the end-user data from being compromised by a malicious ISP (ISP-B).

Aiming to highlight a selected subset of the functionalities supported by our developed security architecture, we next present the architecture and the implementation of our Proof-of-Concept demonstration. Section 11.6 evaluates how the following architecture fulfils these episodes.

11.4 Encrypted SFC Architecture

In this section, we describe the architectural components and the network topology for enabling encrypted and isolated VLs. This work follows the design guidelines from our previous work [3] where we presented an architecture consisting of a tiered structure of data plane and control plane components. This architectural section summarises this work and focuses on the implementation-specific ele-

ments of the design. The main objective of the design is to structure a layered networking architecture into a data centre environment. Specifically, this includes a design of interconnected Compute Node components that are capable of forwarding encrypted SFC packets by the use of two layers of NSH headers. Accordingly, we have provided three data plane components running on the Compute Nodes (Figure 11.3); a new Service Function Forwarder (SFF), a new Encryption Function (EF) and a new forwarding framework for the Service Functions (SF). These components are based on the programmable switch language P4 [9]. Figure 11.3 shows that these data plane components also have their corresponding control plane units, following the Software-Defined Networking paradigm of centralised control and network programmability. We used a micro-service design principle and implemented each of these components as Virtual Machines (VMs). According to our previous work [3], we used RESTConf web services to exchange messages between these components.

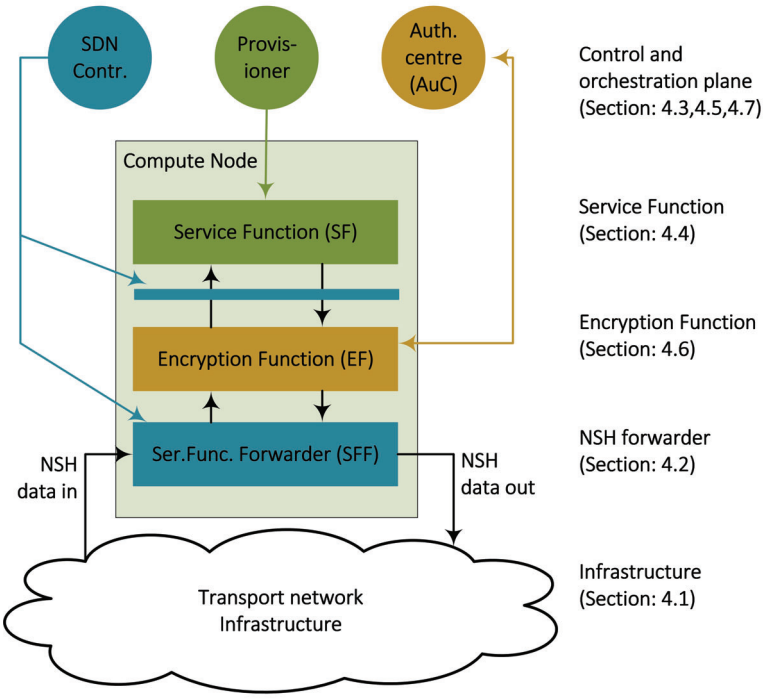


Figure 11.3: Top-level architecture.

The following subsections discuss the functionalities of these data plane compon-

ents (Sections 11.4.2, 11.4.4 and 11.4.6), the infrastructure that they are connected to (Section 11.4.1) and the control plane units they interact with (Sections 11.4.3, 11.4.5 and 11.4.7).

11.4.1 The Infrastructure

The nature of an SFC accommodates Segment Routing (SR) [29, 18, 30], which implies that the sender of an IP packet specifies the packet path. Specifically, SR implies that the packet header contains SFC state information of how to route a specific packet for a selection of intermediate routers. We use NSH as a data plane enabler for SR in order to steer the traffic in such SR paths between the SFs and the EFs (Section 11.4.1). Two layers of NSH headers constitute two overlay networks. One layer addressing the communication between the SFs and one additional layer addressing the point to point communication between the EFs (Figure 11.4).

Currently, SR by NSH is not widely supported by routers and neither is the new NSH encryption header extension that we have suggested. Therefore, in order to ensure packet forwarding through legacy network devices, we define that the NSH packet must be encapsulated by an outer transport network between the NSH-aware routers. Figure 11.4 shows that we use VXLAN-GPE for this underlying network. Each of these network layers accommodates the different communication layers in the architecture. For example, an NSH header is only valid between two SFs, while the new additional NSH header is only valid between two EFs. Hence, the structured packet header (Figure 11.4) is also reflected in a structured design of the networking components (Figure 11.3), where each data plane component is responsible for each layer.

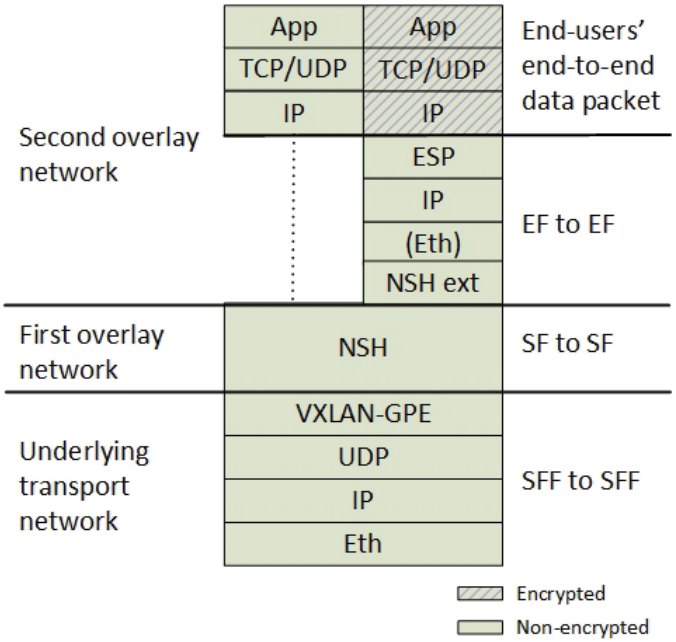


Figure 11.4: The layered network architecture.

Overlay Network Topologies

This structured setup of the networking components ensures that the routing of the data packets is not only controlled by flow-rules, but it is also controlled by how the network topology is designed. This is the main objective behind the design of the structured hierarchy of the NSH headers. The structured network topology disallows unencrypted data traffic between an SF and an EF to be routed out of the Compute Node and out on the network.

Figure 11.5 shows the main difference between using and not using an additional NSH header. Within the current NSH RFC [17] (no additional NSH header), the EF must be treated as a regular SF on one NSH layer (Figure 11.5 (1, 2)). The VL is perceived as encrypted and protected if both the EF and SF is located on the same Compute Node. However, if the EF is migrated to another host (Figure 11.5 (2)), the non-encrypted traffic (between SF-A and EF-X) is in fact allowed to flow both between different Compute Nodes or between different infrastructure domains. Hence, enabling the EF in a separate network layer (Figure 11.5 (3)) makes the network topology more secure. Using two NSH layers ensures that the SF never can be distributed in a way where non-encrypted traffic can leave

the Compute Node. The VM, such as the EF-X (Figure 11.5 (3)), is in this case also open for VM migration, but, if the VM EF-X is migrated to another Compute Node, the next-hop network destination is unavailable due to a header mismatch between the two types of NSH headers. Hence, the VM's EF-X and SF-A must be migrated in pairs for allowing the communication between them.

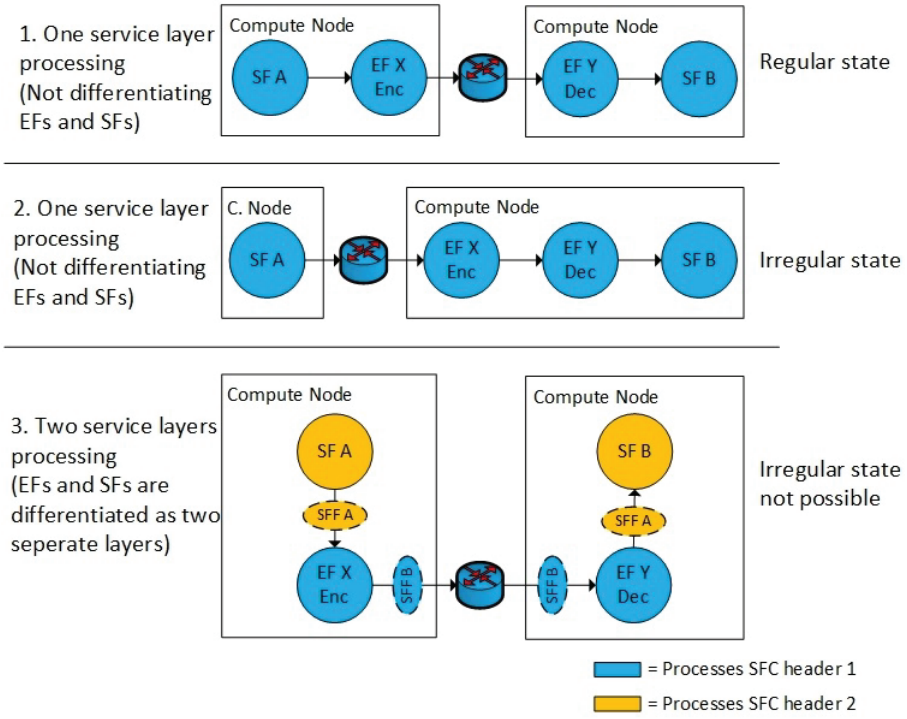


Figure 11.5: An additional encryption overlay.

Dedicated networking components per Compute Node that are responsible for the inner and the outer NSH headers logically separate the EFs and the SFs. Therefore, we implemented two separate virtual switches on each compute node that is responsible for the routing of the two NSH layers. This structured setup of Compute Node components (Figure 11.3) ensures that the EFs must be co-located with the SF.

Underlay Network

A threat to this structured networking model is an underlying VXLAN-GPE network. In traditional data centres with no NSH overlay, VXLAN is not defined as an underlay network, but it constitutes an overlay network by abstracting the phys-

ical network into one big distributed virtualised switch. This implies that, if all the components in our architecture are running as VMs on one underlying distributed virtual switch, the overlaying NSH switches are unaware of the underlying Compute Node location. This compromises the structured setup of networking components on the Compute Node. We solved this problem by combining VXLAN and NSH networks into one customised virtual switch. In addition, we adopted the architectural networking principles from VMware NSX [23] and pinned the virtual switch to the Compute Node and perceived it as a hypervisor component. We simulated this structure by locking the virtual switch VM to the Compute Node and pretending that the virtual switch was not available in the hypervisor user space. According to the SFC RFC [2], introducing NSH/SFC-awareness to a virtual switch makes it a Service Function Forwarder (SFF). Hence, we defined two underlying SFFs to be responsible for each NSH layer respectively and let them also to be responsible for handling the VXLAN-GPE tunnels.

11.4.2 The Service Function Forwarders

In order to support the new NSH packet header formats [3], we created a new customised virtual switch, based on P4. The programming language P4 enables programmers to customise packet forwarding rules in switches. We made a very simple switch that constitutes an SFF, with VXLAN-GPE and NSH forwarding support. The SFF has the following functionality:

- It can parse the new encryption attributes in the NSH header (MD-type = 3, E-SPI, E-SI [3]).
- It classifies IPv4 traffic in order to apply NSH headers.
- The SFF can act as a forwarder for NSH packet destined to other SFFs.
- It can act as an NSH packet forwarder inside an SF in order to make the SF NSH-aware.
- It can provide Layer 2 mac-address resolution based NSH packets instead of using IP and ARP (see Section 11.4.4).
- It can provide VXLAN-GPE support.

11.4.3 The Service Function Forwarder Controller

In our previous work [3], we have suggested using BGP as a control plane mechanism in order to have a standardised method of exchanging NSH routes. However, we identified concerns related to scalability and security of using BGP. Applying route security in BGP includes that each route has to be authorised on a per-peer basis and all viable routes need to be pre-enumerated. In addition, with no route aggregation, route propagation and exponential growth of BGP routes for EFs, we question the scalability of BGP as an NSH control plane protocol. Hence, we

changed the control plane component from our previous work [3] from BGP to distributed RESTconf. This opened up for more efficiently giving the exact routing instructions to the specific Compute Nodes only. RESTconf also enabled a more flexible authorisation of the NSH routes, by authenticating the RESTconf connection by the Secure Socket Layer (SSL).

However, we also selected RESTconf as the control plane protocol due to interoperability reasons. In federated multi-tenant NFV environments, an overlay network is created with virtual forwarding devices such as an SFF. This opens up for customising the virtual network devices and the network controller. This enables network operators to deploy customised networking software in an agile and fast manner. By utilising P4, it is also possible to specify customised flow rules when configuring these network devices. Hence, we utilised the feature of SDN-based network programmability, by specifying customised network configuration by the use of P4 flow rules over RESTconf. The need for customised flow rules is reasoned by the new NSH header extension.

For Proof-of-Concept purposes, we created a very simple RESTconf based network controller with a set of predefined P4 flow-entries. It consists of a simple HTTP client that distributes flow rules over RESTconf by using Linux shell scripts.

11.4.4 Service Functions

The Service Functions (SFs), applied as VMs, are intended to manipulate the end-user data packets. We simulated that we used SFs acting as a vCPE, video caching service and a virtual firewall by using dummy services. Therefore, all SFs are configured to simply forward all data traffic according to the flow-specification rules we apply. The SFC RFC 7665 [2] defines that an SF has primarily two data plane interfaces: one for incoming and one for outgoing traffic. Inside the SF, the packet forwarding is explicitly set to follow the SFC directions and not the standard routing table. Specifically, data traffic coming in on one interface must go out on the other interface and vice versa. We solved this SF routing problem by making the SF NSH-aware. This implies that the NSH header is not removed when a packet enters an SF. Due to the lack of NSH state capabilities in operating systems such as native Linux systems, we introduce a new virtual NSH network stack inside the SF. This new network stack is an NSH-aware P4 switch which acts as a front-end network stack inside the SF. This principle of NSH-awareness in the SF is extracted from the VXLAN-tool [31] implementation and adopted to a P4 environment in order to support the new NSH header. We defined the following features in the SF:

- One SF can appear multiple times in one single SFC. Hence, the SF is NSH-

aware by using an underlying P4 switch in the SF. The P4 switch is connected to two virtual veth interfaces facing the SF application and two native interfaces facing network interfaces of the VM.

- According to the SFC specification, the SF should be independent of the IP subnet topology between the SFs. This means that the IP subnets connected to VM interfaces do not follow standard IP subnetting topologies. For example, when an SFC changes, the mac-address of the next-hop SF are also changing. From an SF perspective, this mac-address has to correspond to the next NSH hop. Hence, the virtual P4 switch in the SF must be able to map interface mac-addresses to SFs. For outgoing traffic from an SF, we use dummy static destination mac-addresses. For incoming traffic to an SF, it is the responsibility of the P4 switch to set the correct destination mac-address to the IP interface of the SF. This mac-address is based on next hop in the SFC. Hence, instead of using standard ARP as a binding between layer 2 and layer 3, we introduce a new mac-address mapping scheme between mac-addresses and NSH Service Function Identifiers. This is implemented as P4 flow-rule actions. A dynamic side effect of this is that the IPv4 addresses of the SF application theoretically can be reused for each hop in an SFC.

11.4.5 Service Function Provisioner

The Service Function Provisioner component corresponds to the Virtual Infrastructure Manager (VIM) in the NFV reference model [32]. It is responsible for maintaining the lifecycle management of all virtual network functions. We simplified this function and used Vagrant [33] and Vagrant scripts as a provisioning tool for all VMs per Compute Node. As an overlay to multiple Vagrant nodes, we used RESTconf to instantiate the Vagrant scrips.

11.4.6 The Encryption Service Function

This component is responsible for both encrypting and decrypting the data traffic in front of the SF. This functionality is realised with a data encryption application in a customised SF that we named the Encryption Service Function (EF). From a network infrastructure perspective, the EF is a copy of the SF, except for being responsible for a different network layer (the additional NSH layer). In addition to the P4 networking functionalities, the EF adopts the Software-Defined IPsec application (SD-SA) functionality that we have presented in our previous work [7]. In summary, this application has the following features:

- We use the Linux based IP XFRM application to encrypt and decrypt IP packets and encapsulates them with an IPsec Encapsulating Security Payload (ESP) header.

- The encryption application runs inside a Linux network namespace (netns) that separates the encryption application from the P4 switch.
- IPsec Internet Key Exchange (IKE) is replaced with a new web service application that exchanges the encryption keys and the integrity keys in a separate control plane channel.
- The EF is instantiated with a set of preshared keys. These keys are used to establish a secure connection to a centralised Authentication Center (AuC) that manages the key distribution.

11.4.7 The Authentication Centre (AuC)

The EFs are controlled by an Authentication centre that distributes the encryption keys and the integrity keys. Due to the non-bidirectional NSH communication channel between EFs [2], an IPsec IKE channel is not possible to establish on the data plane. Hence, we adopt the aforementioned SD-SA application from our previous work [7] in order to replace IKE in IPsec. In summary, this application includes the following functionality. The initial step is to pre-configure an authentication key for every EF during EF instantiation. Second, all EFs establish a secure channel to the AuC. Third, the AuC sends the integrity and confidentiality key to the encryption function over the authenticated and secured RESTconf channel. This last step is a periodic event that is repeated for every key change. An important requirement for this concept to work is that all EFs are connected to one common AuC. This also requires a shared control plane VPN between all data centre tenants. This control plane VPN is established by using site-to-site IPsec VPN tunnels between the data centres.

11.5 Implementation

Based on the aforementioned scenarios (Section 11.3), we constructed a network topology consisting of three simple SFs and two underlying pairs of encrypted channels. Figure 11.6 shows the components that are involved in the data plane forwarding of the NSH packets.

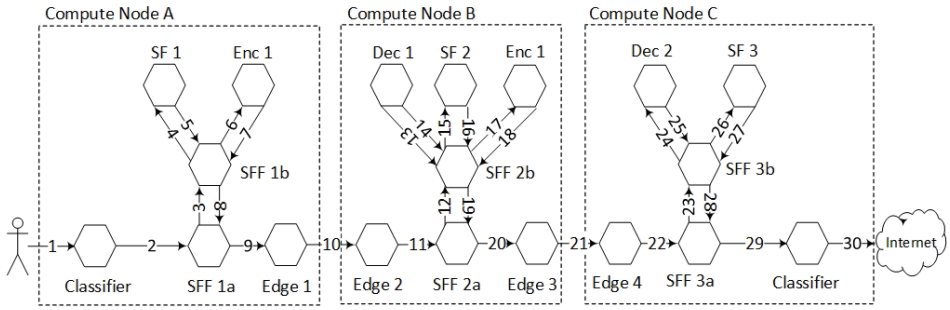


Figure 11.6: Service plane topology.

This network topology setup implements the use case scenario (Figure 11.2) and also reflects the tiered network topology (Figure 11.3). We implemented this topology by running one Compute Node per ISP where each Compute Node had one inner and one outer SFF. We also configured an Edge gateway per network domain that was responsible for interconnecting the data centre domains. We used one Compute Node per NFV domain. Correspondingly, there is one Edge gateway per Compute Node.

Figure 11.7 shows the categorisation, the enumerations and the virtual bridge connections of the VMs running on each Compute Node. The SFs, the EFs and the classifiers are instantiated as multiple instances of VMs. These VMs are instantiated per SFC during service provisioning. The SFFs are statically deployed VMs that are pinned to the hypervisor. The control plane components are also categorised as a special group of VMs. This is because they are only instantiated at one of the Compute Nodes and because they are not connected to the data plane.

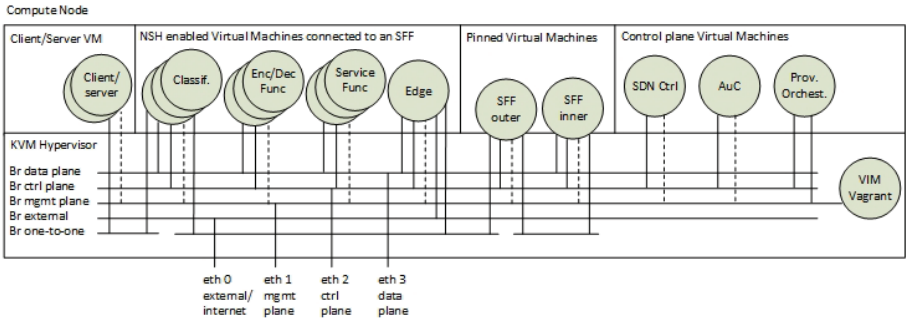


Figure 11.7: Virtual Machines and networks per Compute Node.

From a networking perspective, Figure 11.7 also shows that each VM is connec-

ted to different Linux-bridge domains. We used virtual Linux interfaces (veth) to interconnect VMs to virtual bridges. Furthermore, these virtual bridges are also connected to the physical network interfaces. This network construction follows the principle of virtual network infrastructures in Linux that is also used in, for example, OPNFV [34] and OpenStack [35].

For local Virtual Infrastructure Management (VIM), we ran the VM provisioning tools and the local network/bridge management as non-virtual function alongside the Kernel-based Virtual Machine (KVM) environment. We used Linux scripts and Vagrant to control the instantiation of VMs and to control the mapping the virtual network interfaces to the underlying VXLAN-GPE infrastructure. The local VIM is orchestrated by a simple top-level RESTconf based orchestrator. Figure 11.8 shows the hierarchy of both this orchestrator and the other control plane components. For proof-of-concept purposes, we only used one Compute Node per domain controller.

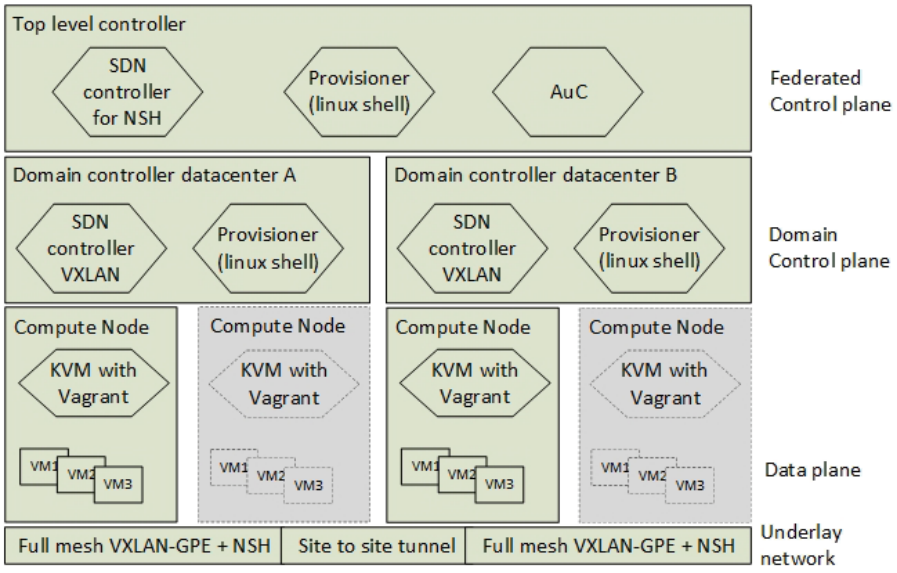


Figure 11.8: Hierarchy of network control.

The NFV implementation is developed by the use of Vagrant, Linux bash scripting and the switch programming language P4. The source code, the demonstrations and the test results are available at `https://github.com/gunleifsen/encNSHinP4`.

Lab Setup and Tools

We set up our experiment by using four HP Proliant DL380 G7 servers. All servers had two 3.47 GHz Intel Xeon X5690 processors with six cores each, 196 GB of DDR3 memory and 1 Gbit network adaptors (In a testbed provided by Eidsiva broadband, Oslo, Norway). The servers ran Ubuntu 18.04.2 LTS with kernel 4.15.0-47-generic. We used Vagrant 2.2.4 and qemu-kvm 1:2.11 with virtual network adapters by libvirt. For every Compute Node, we disabled Hyper-Threading, Turbo Boost, and power saving.

We simplified the instantiation of the VMs by only using one VM template for every VM. This VM template was set up with Ubuntu Linux 18.04.2 LTS, 2 GB RAM and one virtual CPU. The pre-installation of the template included the P4 framework (`https://github.com/jafingerhut/p4-guide`) where we used the P4 version P4_16 with the behavioural model 2 (BMv2) from Barefoot [36]. For running the P4 code, we used the inner virtualisation software from Barefoot named the "simple_switch".

For all communication between the VMs, we simulated RESTconf by the use of simple web services running over secure netcat (socat) [37].

According to the scenario and the service plane topology, we created 21 VMs for data plane forwarding, including two endpoints (Figure 11.6). In addition, for the federated top-level controller, we combined the network controller, the service provisioner and the AuC in one common VM (22 VMs in total).

For end-to-end traffic testing, we used Iperf 3.0.11 for measuring the performance and provisioning time. By sending a fixed stream of packets per seconds, we measured the network outage time by counting the packet loss. For packet injection tests, we used the Python tool scapy [38] and, for packet monitoring, we used Tcpdump on the virtual Compute Node interfaces (veth).

11.6 Verification and Results

This section presents the results of the three verification episodes we introduced in Section 11.3. This includes (1) a packet forwarding and provisioning episode, (2) a resilience and availability episode and (3) a security integrity episode.

11.6.1 Episode 1: Packet Forwarding and Provisioning (req: i, iii, iv, vi)

This episode aims to provide a demonstration of the service provisioning. It also aims to show that the data packets are routed correctly and that end-to-end traffic tests and throughput tests are satisfying the requirements.

Service Provisioning Times

From an end-user perspective, it is expected that their NFV ISP provides them with on-demand service provisioning and a secure infrastructure. We developed episode 1 in order to provide a Proof-of-Concept demonstration of the implemented architecture with respect to service provisioning. This demonstration also shows that the layered NSH header architecture is capable of forwarding encrypted SFCs according to a network forwarding policy in a relatively fast and reliable manner. Specifically, we developed a test that demonstrates that the designed architecture is able to set up an SFC according to the requirement of hop-by-hop encryption with a new NSH header (req: i, iii) and by the use of RESTconf flow distribution (req: iv). We provide these demonstrations by monitoring the provisioning time of a subset of the services in the architecture.

The creation of VMs, the altering of the routing paths and the reestablishment of the encryption keys, result in network outage from an end-user perspective. Hence, we measured this time by monitoring network outage during service provisioning. We set up a simple Iperf measuring test between the client and server and experimented with altering the SFC. We altered the SFC from SF1 $->$ SF3 to SF1 $->$ SF2 $->$ SF3. This simulated turning on and off the SF located at ISP-B and implicitly adding and removing a service from the SFC.

By sending a fixed stream of 64 packets per seconds from the client to the server, we measured the network provisioning time by monitoring the packet loss during these service alternations. One packet loss of 64 packets corresponds to approximately 15,63 ms of a network outage. The measurements were aiming to measure seconds. Therefore, we set the packet rate at 64 pps. This also ensured a minimisation of a potential packet loss due to unrelated reasons such as collisions, traffic congestion or buffers overflows.

We differentiated the measurements (Table 11.1) in three types. Full provisioning (1a) with no traffic flow during provisioning, Soft provisioning (1b) where the data traffic runs while new services are instantiating and (1c) we also measured the periodic key change provisioning times.

Setting up an additional SF with encrypted links includes; (1.a-1) instantiating the SF VM, (1.a-2) instantiating two EF VMs, (1.a-3) authenticating the EFs and distribute the encryption keys and (1.a-4) distribute the new flow routes. Hence, we measured the provisioning time for each of the sub-processes and summarized the total provisioning time (1.a).

The instantiation of the SFs (1.a-1) and the EFs (1.a-2) are the most time-consuming processes. However, these processes can be instantiated before the traffic flows are

redirected into the new VMs. This also includes the setup and authentication of the EFs (1.a-3). We refer to this process as soft provisioning (1.b). It is expected that such a planned provisioning is most common. Here, the distribution of the flow rules is not assigned before the VMs are fully provisioned and EFs are authenticated. This scheduling of network provisioning significantly decreases the network outage time. However, for non-controlled events, such as hardware failure, the re-provisioning time increases due to the failure detection time and due to the lack of pre-instantiated VMs (see Section 11.6.2).

The periodic key change provisioning time is the most frequent provisioning process. It is assumed that the encryption keys are set up to change once every hour. However, for measurement purposes, we set the periodic key change interval to 5 s. Our previous work [7] indicated that the periodic key change provisioning had limited effects on the network outage time. This is also confirmed by these measurements where the network outage time for each key change is about 0.1 s (1c).

From an end-user's perspective, the most relevant network outage times are the outages during the soft provisioning and during periodic key changes. It is expected that most web applications based on TCP, such as Youtube and Netflix, are resistant to these network outage times that are less than 100 ms. Re-using an EF in order to reconnect it to a different EF peer has a provisioning time of 1.2 s (1.b-1). Hence, this result indicates that it is more efficient to pre-instantiate and pre-authenticate a new pair of EFs instead of reusing any existing EFs during re-provisioning. This pre-authentication consequently sets the soft provisioning time and the network outage time to only include the time it takes to distribute the flow rules. It is assumed that the network outage time caused by the re-distribution of the flow rules is not perceived as network outage by most end-users.

Table 11.1: Provisioning times.

Episode	Test Name	Packet Rate	Packets Lost	Outage Time
1.a	**Full provisioning**	64 pps	13,184	206.0 s
1.a-1	-SF instantiation	64 pps	3049	47.6 s
1.a-2	-EF instantiation	64 pps	10,176	159.0 s
1.a-3	-Auth and encr. setup	64 pps	81	1.2 s
1.a-4	-Distribution of flow rules	64 pps	0	0.0 s
1.b	**Soft provisioning**	64 pps	154	1.2 s
1.b-1	-Auth and encr. setup	64 pps	81	1.2 s
1.b-2	-Distribution of flow rules	64 pps	0	0.0 s
1.c	**Periodic keychange (every 5 s)**	64 pps	7	0.1 s
2.a	**Failover with protection**	64 pps	326	5.0 s
2.a-1	-Detection time			5.0 s
2.a-2	-Distribution of flow rules	64 pps	0	0.0 s
2.b	**Failover without protection**	64 pps	13,504	211.0 s
2.b-1	-Detection time			5.0 s
2.b-2	-Full provisioning	64 pps	13,184	206.0 s

pps = packet per second.

Throughput

According to the requirement of adoption (req: vi), we argue that the architecture fulfils this requirement by using fully virtualised overlay networks. The new NSH header only needs to be implemented in virtualised environments and therefore it is also easily deployed in fully isolated, autonomous and customer-specific environments. However, other important factors for adopting the architecture are scalability and throughput performance. The TCP throughput is a product of bit rate, packet-size and latency. Hence, the latency factor for throughput performance is highly dependent on the number of NSH forwarders, their latency in processing packets and the latency between them. Hence, we measured the throughput by varying the number of SF and EF hops (Figure 11.9). We tested the throughput from the client to the server by using an Iperf TCP bandwidth test with window-size 512.

The results (Figure 11.9) show a decreasing throughput when the number of SF hops increases. The main reason behind this result is the increased latency that the virtual machines introduce. We measured that a VM with a P4 enabled switch in average used 6 ms to process a packet. The virtualisation software we used for P4

is based on CPU processing without any network accelerator driver. This lack of suitable software drivers for P4 explains this latency. However, it is expected that new generations of OVS, IPAC and other P4 runtime environments will decrease the packet processing latency in future releases of the P4 runtime frameworks.

However, the most important factor with respect to NFV adoptability (req: vi) is how much degradation in throughput the encrypted links introduce. Hence, we measured the difference between EF and SF hops, by creating multiple SFCs with and without encrypted Virtual Links. The graphs (Figure 11.9) show that there is no significant difference in the throughput between an SFC with EFs only and an SFC with SFs only. This confirms that it is the P4 switch that introduces the latency and that the performance degradation is due to the P4 hypervisor or the effectiveness of the P4 program.

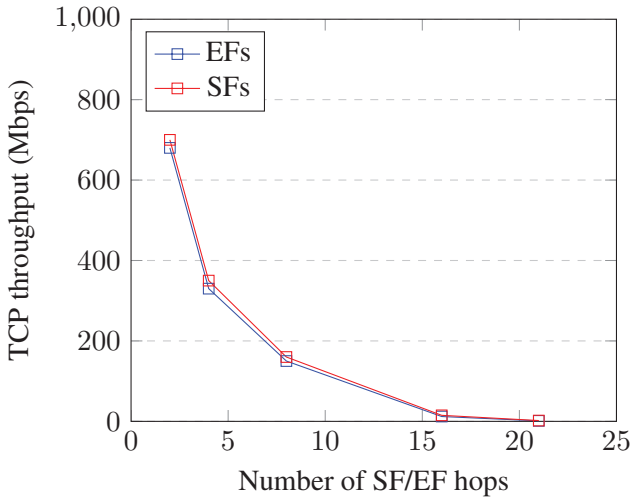


Figure 11.9: TCP throughput per SF/EF hop.

Summary

The overall results show that the solution is fulfilling the following requirements:
Req i ✓: We have shown that the architecture provides hop-by-hop encryption by monitoring the traffic flows before and after the provisioning of the EFs. In addition, small network outages during re-keying confirm that the encrypted links are running and that the key change is working.
Req iii ✓: The fact that requirement i is fulfilled also confirms that the forwarding of the new packet header is possible when the Virtual Links are encrypted. Hence, the new SFC header is not encrypted. This is also confirmed by analysing SFC

packets.
Req iv ✓: By running provisioning and alternating different network topologies, we have shown that a RESTconf based control plane in a multi-tenant environment is capable of distributing micro-segmented flow rules according to the requirements.
Req vi ✗: By applying the NSH header in an interconnected multi-tenant data centre, we have shown that the new NSH header is easily adaptable (Section 11.4.1). This is because the NSH header is applied in an overlay network. Throughput tests have shown that a single EF does not degrade the throughput performance more than a single SF. However, due to the lack of an effective P4 virtualisation platform in our implementation, the performance significantly degrades when the number of SFs increase. Hence, the adoption of P4 in NFV is highly dependent on hardware accelerators.

11.6.2 Episode 2: Resilience and Availability (req: v, vii)

The capacity of the architecture to adapt to controlled network alterations has been demonstrated earlier with respect to the network provisioning. This episode has been developed in order to highlight the details of the process and to show how the architecture responds to software or hardware failures (req: vii). Hence, we simulated that the intermediate ISP-B has a hardware failure and becomes unavailable. In addition, we have also simulated the consequences of unavailable control plane components and tested protection mechanisms in the key distribution method (req: v).

SFC Protection

Our Proof-of-Concept implementation of the P4 data plane did not include any event handlers of detecting network failures such as link-down or node-down (such as LLDP discovery events in OpenFlow). However, we simulated an "NFVi fault and management" [1] approach similar to ceilometer monitoring in OpenStack, by monitoring the VMs instead of the virtual switch interfaces. Consequently, we assume that the NFV infrastructure is capable of automatically migrating the VMs to another Compute Node during hardware failures. However, this migration process results in network outage time for the end-user where network outage time = detection time + VM provisioning + key distribution + route distribution (Table 11.1 (2.b)). In order to reduce the network outage time, it is possible to pre-instantiate redundant VMs. Consequently, the VM provisioning time is removed from the equation. Hence, we simulated a protection of software and hardware failure by duplicating the VMs running at ISP-B to also run at ISP-A. By letting the VMs run in both domains, the VM becomes protected and, consequently, it also creates an SFC protection. In order to simulate a hardware failure, we manually

shut down the VMs running at ISP-B by using KVM virsh directly on the VMs.

Table 11.1 (2.a, 2.b) shows the failover time when SFC protection is enabled and when it is not. A network topology with SFC protection enabled provides evidence for a resilient and dynamic SFC architecture (req: vii). However, it is noted that the SFC protection introduces an additional resource consumption for all the VMs that are running in standby mode.

A method of reducing the resource consumption is to reduce the number of duplicate EFs. The aforementioned demonstration duplicated four EFs. Consequently, the additional pairs of EFs were already authenticated and failover is performed by a distribution of the flow rules only. By not duplicating EF1 and EF4, these EFs can be reused in order to reconnect to the duplicated versions of EF2 and EF3. This adds an additional 1.2 s to failover process, but it reduces the overall resource consumption.

Protection of Invalid VM Migration

The main objective of the architecture is to enable hop-by-hop encryption of VLs (req: i). This includes protecting the data traffic between two SFs with a pair of EFs. Hence, one of the most important features of the architecture is to ensure that non-encrypted data between an SF and an EF reside on the same Compute Node. We demonstrated the feasibility of the hop-by-hop encryption during the provisioning in episode 1 by running end-to-end traffic tests. Here, we also verified that the traffic was encrypted by monitoring the data traffic in the SFFs.

However, with respect to resilience, the aforementioned episode did not consider the consequences of VM migration. A failover of the VMs implies that SFs and their connected EFs must be migrated all together. A misconfiguration or failure in the VM migrations can result in an irregular topology of SFs and EFs. Hence, we aim to verify that the layered network topology introduced in Section 11.4.1 is protecting the non-encrypted flows from entering the network. We configured an invalid network topology by migrating one of the EFs to another Compute Node as we showed in Figure 11.5 (2). In this case, our architecture and flow policy disallowed the distribution of the flow rules. This result was expected. However, we successfully managed to manually override and manipulate the flow rules in order to allow such traffic flows. In order to accomplish this, we had to (1) manipulate the NSH packet coming from the SF and (2) define a flow rule that sets the NSH next hop to be a remote SFF destination for an NSH packet that is tagged for going into an EF for encryption. This is a clear flow-rule policy violation. This policy violation is easy to detect because the SFFs are pinned to the Compute Node. Hence, an NSH packet that is heading to an encryption function should

never leave the outer SFF.

The demonstration confirms that the tiered SFF infrastructure and the pinned SFFs make it easy to control the inner encryption flows. It also shows that there is a need for providing policy rules for the P4 switches. We enforced this rule only by implementing it in the overlay RESTconf API.

Key Distribution

Due to the lack of a direct data plane communication between two SFs, we developed a new key exchange mechanism by using centralised key distribution (req: v). Our previous work [7] has shown that SD-IKE with a centralised key distribution and Authentication centre (AuC) makes key distribution more efficient in non-NFV environments. The SFC alternations in episode 1 and the SFC failover protection in SFC episode 2 (Section 11.6.2) provides a Proof-of-Concept demonstration for integrating this concept in NFV.

Availability of Control Plane Components

The aforementioned key distribution is not only performed during provisioning, while the encryption keys change periodically. This makes the AuC service a critical component. Killing the AuC service does not affect the end-user traffic. It only stops the encryption keys from being renewed. Hence, an additional key monitoring agent must run along with the encryption and decryption services. This monitoring agent detects if a new key is not received within a certain expire time. If the expiry time is reached before a new key is received, the EF deletes its' Security Association.

Instead of shutting down the AuC, we simulated this key protection feature by manipulating the key expire time in the encryption service. We experimented with setting the expire time to 10 s in the encryption service and the AuC re-keying time to 20 s. Next, we ran a simple Iperf bandwidth test for 60 s and used tcpdump to monitor if the packets traversed the intermediate SF. We confirmed the functionality by observing that the EFs periodically stopped working every for 10 s.

This declarative SDN approach for the AuC also applies to the network controller. When the SDN controller has distributed the flow rules to the SFFs, it is expected that the SFFs continue forwarding packets even after the SDN controller becomes unavailable. Consequently, network topology changes and VM migration together with a non-functional SDN controller results in non-functional SFCs. We confirmed this behaviour by shutting down the SDN controller.

Summary

This section has shown that the following requirements are satisfied:
Req vii ✓: First, the architecture opens up for enabling protection of EF and SF. The measurements of failover times provide evidence for a resilient architecture towards hardware or software failure. Second, we have shown that the layered network infrastructure provides resilience towards misconfiguration of VMs. Third, a security feature of protecting network security during AuC or network controller outage have been demonstrated.
Req v ✓: SFC alternation in Episode 1 and the protected failovers in Episode 2 provides evidence for an effective key distribution method in NFV environments.

11.6.3 Episode 3: Security Integrity (req: ii, viii)

For this episode, we simulated that the intermediate ISP (ISP-B) is compromised. The objective of the demonstration is to show that the integrity of the architecture is maintained for ISP-A and ISP-C even if ISP-B is compromised. We demonstrate this by showing that the architecture supports flow-based encryption (req: ii) and by showing that it is resistant to manipulation of the packet headers or flow injections (req: viii).

Eavesdropping

In order to demonstrate flow-based isolation and encryption (req: ii), we defined one SFC with multiple inner encrypted flows. The SFC is defined as SF1 $->$ SF2 $->$ SF3 where the flows are defined as email traffic and video traffic (over HTTP). SF1 is a vCPE that contains multiple network services. However, in this episode of the scenario, we define it to handle email traffic. Hence, SF1 is set up to handle all email traffic (port 25,110), while SF2 handles video traffic over HTTP (port 80,443). SF3 is a firewall that handles both flows. Due to the risk of eavesdropping, we encrypt the email traffic between SF1 and SF3 and we encrypt the HTTP traffic for SF1 $->$ SF2 and SF2 $->$ SF3.

We ran both traffic types simultaneously by running two instances of Iperf on two different ports, namely flow 1 (port 25) and flow 2 (port 80) (Figure 11.10). Hence, we classified the traffic into two different types, in total, three pairs of encrypted links. Flow 1 was defined as SF1 (vCPE) $->$ EF1 $->$ EF2 $->$ SF3 (vFW). Flow 2 was defined as SF1 (vCPE) $->$ EF3 $->$ EF4 $->$ SF2 (vVideo) $->$ EF5 $->$ EF6 $->$ SF3 (vFW). By using tcpdump, we observed both encrypted and non-encrypted traffic flows at ISP-B. This confirmed that the architecture was supporting flow-based encryption.

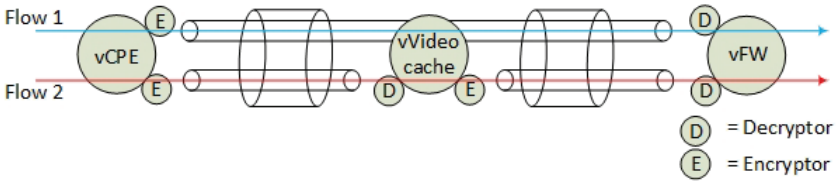


Figure 11.10: Flow-based encryption test in episode 3.

Route and Packet Injection

The main focus of our contribution was to protect the SFC from being eavesdropped when the data traffic is traversing ISP-B. However, another attack method of gaining access to unauthorised traffic is to manipulate the packet forwarding services in order to redirect the traffic path and implicitly gain access to non-encrypted traffic. Hence, an important feature of the architecture is that it is resistant to packet injection or route injection (req: viii).

We did not find any method to inject flow rules into the P4 switch without compromising the network controller. However, for test purposes, we turned off the authentication and SSL on the RESTconf interface towards to P4 switch. This allowed us to manipulate the flow rules and consequently also manipulate the SFC.

For packet injection, we simulated that SF2 is compromised in ISP-B. We used an NSH extension in the Python based scapy tool [38] in order to inject packets into the data plane. We successfully managed to inject packets from an SF to another SF that was belonging to another SFC. Due to a lack of security features in the SFF, the SFF was not able to detect the VM source of the injected packets and simply forwarded the packet according to the packet headers. However, this was only possible for non-encrypted data flows.

This packet injection problem is similar to spoofing the source address field in IP packets. A possible solution to this problem is to introduce an integrity attribute in the NSH headers. Extending the NSH integrity header RFC [39] to support layered NSH can potentially ensure that the NSH packet originates from a valid VM source.

Summary

The verification of the security integrity objectives is summarised as follows:
Req ii ✓: We confirmed that the architecture supports flow-based encryption by observing encrypted and non-encrypted traffic traversing ISP-B. End-to-end traffic tests also confirmed that each flow was tagged with two different types of inner NSH encryption headers.

Req viii ✓: The architecture is resistant to simple packet injections and to the manipulation of flow rules. However, this packet integrity is established by using IPsec over the encrypted links. Hence, it requires that every Virtual Link is encrypted and that the Compute Nodes are not compromised. A possible solution in order to ensure packet integrity to non-encrypted data are to add an integrity key to the NSH header.

11.7 Conclusions

This paper proposes an architecture for on-demand provisioning of encrypted and isolated SFC using P4, NFV and SDN architectural principles.

A comprehensive view of the developed security framework for SFC has been presented, according to the scenarios executed during the concluding system validation demonstrations. A subset of the security-related functionalities supported by the developed architecture has also been shown in order to highlight critical architectural details towards its implementation.

Furthermore, this article unifies the publicly available results of our security-related studies, by highlighting how the distinct components presented earlier interoperate towards providing secure SFCs. The presented results highlight the capacity of micro-segmented SFC in NFV, given that the corresponding security requirements are satisfied.

The presented architecture is based on virtualised overlay networks and the upcoming technology P4. These technologies aim to overcome network protocol standardisation and interoperability issues. Hence, the architecture is applicable in any IP network and any Infrastructure as a Service (IaaS) platform. However, this abstraction of physical resources raises new standardisation issues within the virtualised environment, such as the encryption application in the service function. This puts a burden on the SF developers and calls for a standardisation of the encryption application interfaces in the SF and the AuC. Hence, this proof of concept experiment aims to contribute to the standardisation of NFV application interfaces for enabling encrypted Virtual Links.

Through the executed studies of encrypted SFCs, a variety of future work paths have been identified. These include the investigation of hardware accelerators, integrated QoS, availability and security policies, particularly for protected and encrypted SFC. Furthermore, another potentially critical path of future work refers to the investigation of packed injection between SFCs where encryption enabled SF is a possible solution.

Author Contributions:

The main author of this paper is H.G.; V.G. and T.K. have contributed with respect to the paper structure, quality assurance and editing.

Funding:

This research was funded by Eidsiva, the Norwegian Research Council and the Norwegian University of Science and Technology (NTNU).

Conflicts of Interest:

The authors declare no conflict of interest.

References

[1] ETSI. *Network Function Virtualization (NFV) Architectural Framework v1.1.1*. Available online: http://www.etsi.org/deliver/etsi_gs/NFV/001_ 099 /002/01.01.01_60/gs_NFV002v010101p.pdf (accessed on 23 August 2019). 2014.

[2] Joel M. Halpern and Carlos Pignataro. *Service Function Chaining (SFC) Architecture*. RFC 7665. Oct. 2015.

[3] Håkon Gunleifsen, Vasileios Gkioulos and Thomas Kemmerich. 'A Tiered Control Plane Model for Service Function Chaining Isolation'. In: *Future Internet* 10.6 (2018), p. 46.

[4] Håkon Gunleifsen and Thomas Kemmerich. 'Security requirements for service function chaining isolation and encryption'. In: *IEEE 17th International Conference on Communication Technology (ICCT)*. 2017, pp. 1360–1365.

[5] Ken Peffers et al. 'A design science research methodology for information systems research'. In: *Journal of management information systems* 24.3 (2007), pp. 45–77.

[6] Håkon Gunleifsen, Thomas Kemmerich and Slobodan Petrovic. 'An End-to-End Security Model of Inter-Domain Communication in Network Function Virtualization'. In: *Norsk Informasjonssikkerhetskonferanse (NISK): Bergen, Norway* (2016), pp. 7–18.

[7] Haakon Gunleifsen, Thomas Kemmerich and Vasileios Gkioulos. 'Dynamic setup of IPsec VPNs in service function chaining'. In: *Computer Networks* 160 (2019), pp. 77–91.

[8] Clarence Filsfils et al. *Segment Routing Architecture*. RFC 8402. July 2018.

[9] Henning Stubbe. 'P4 compiler & interpreter: A survey'. In: *In Proceedings of the Seminars Future Internet (FI) and Innovative Internet Technologies and Mobile Communication (IITM), Munich, Germany*. Vol. 47. 2017, pp. 1–72.

[10] Ahmed M Alwakeel, Abdulrahman K Alnaim and Eduardo B Fernandez. 'A Survey of Network Function Virtualization Security'. In: *SoutheastCon 2018, St. Petersburg, FL, USA*. IEEE. 2018, pp. 1–8.

[11] Ibrahim Afolabi et al. 'Network slicing and softwarization: A survey on principles, enabling technologies, and solutions'. In: *IEEE Communications Surveys & Tutorials* 20.3 (2018), pp. 2429–2453.

[12] Mahdi Daghmehchi Firoozjaei et al. 'Security challenges with network functions virtualization'. In: *Future Generation Computer Systems* 67 (2017), pp. 315–324.

[13] Deval Bhamare et al. 'A survey on service function chaining'. In: *Journal of Network and Computer Applications* 75 (2016), pp. 138–155.

[14] Jesse Gross, Ilango Ganga and T. Sridhar. *Geneve: Generic Network Virtualization Encapsulation*. Internet-Draft draft-ietf-nvo3-geneve-13. Work in Progress. Internet Engineering Task Force, Mar. 2019.

[15] Ali Sajassi et al. *Secure EVPN*. Internet-Draft draft-sajassi-bess-secure-evpn-02. Work in Progress. Internet Engineering Task Force, June 2019.

[16] Fabio Maino, Larry Kreeger and Uri Elzur. *Generic Protocol Extension for VXLAN*. Internet-Draft draft-ietf-nvo3-vxlan-gpe-06. Work in Progress. Internet Engineering Task Force, Apr. 2018.

[17] Paul Quinn, Uri Elzur and Carlos Pignataro. *Network Service Header (NSH)*. RFC 8300. Jan. 2018.

[18] Ahmed Bashandy et al. *Segment Routing with MPLS data plane*. Internet-Draft draft-ietf-spring-segment-routing-mpls-22. Work in Progress. Internet Engineering Task Force, May 2019.

[19] Evangelos Haleplidis et al. 'Network programmability with ForCES'. In: *IEEE Communications Surveys & Tutorials* 17.3 (2015), pp. 1423–1440.

[20] Raul Munoz et al. 'Integrated SDN/NFV management and orchestration architecture for dynamic deployment of virtual SDN control instances for virtual tenant networks'. In: *Journal of Optical Communications and Networking* 7.11 (2015), B62–B70.

[21] Hongtao Yin et al. *SDNi: A Message Exchange Protocol for Software Defined Networks (SDNS) across Multiple Domains*. Internet-Draft draft-yin-sdn-sdni-00. Work in Progress. Internet Engineering Task Force, June 2012.

[22] Adrian Farrel et al. *BGP Control Plane for NSH SFC*. Internet-Draft draft-ietf-bess-nsh-bgp-control-plane-11. Work in Progress. Internet Engineering Task Force, May 2019.

[23] Tony Sangha and Bayu Wibowo. *VMware NSX Cookbook*. Packt Publishing Ltd, 2018.

[24] ONF. *Open Networking Foundation, Stratum Project*. Available online: https://www.opennetworking.org/stratum/ (accessed on 04 June 2019). 2019.

[25] Pat Bosshart et al. 'P4: Programming protocol-independent packet processors'. In: *ACM SIGCOMM Computer Communication Review* 44.3 (2014), pp. 87–95.

[26] Rafael Lopez, Gabriel Lopez-Millan and Fernando Pereniguez-Garcia. *Software-Defined Networking (SDN)-based IPsec Flow Protection*. Internet-Draft draft-ietf-i2nsf-sdn-ipsec-flow-protection-04. Work in Progress. Internet Engineering Task Force, Mar. 2019.

[27] David Carrel and Brian Weis. *IPsec Key Exchange using a Controller*. Internet-Draft draft-carrel-ipsecme-controller-ike-01. Work in Progress. Internet Engineering Task Force, Mar. 2019.

[28] Susan Hares. *Use Cases for Resource Pools with Virtual Network Functions (VNFs)*. Internet-Draft draft-hares-vnf-pool-use-case-02. Work in Progress. Internet Engineering Task Force, July 2014.

[29] Clarence Filsfils et al. 'The Segment Routing Architecture'. In: *2015 IEEE Global Communications Conference (Globecom)*. IEEE. 2015, pp. 1–6.

[30] Clarence Filsfils et al. *IPv6 Segment Routing Header (SRH)*. Internet-Draft draft-ietf-6man-segment-routing-header-21. Work in Progress. Internet Engineering Task Force, June 2019.

[31] *The VXLAN-tool website*. Available online: https://github.com/opendaylight/sfc/blob/master/sfc-test/nsh-tools/vxlan_tool.py (accessed on 04 June 2019). 2015.

[32] ETSI. *Network Functions Virtualisation (NFV) Management and Orchestration 001 v1.1.1*. Available online: http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf (accessed on 24 January 2019). 2013.

[33] Michael Peacock. *Creating Development Environments with Vagrant*. Packt Publishing Ltd, 2015.

[34] Amar Kapadia and Nicholas Chase. *Understanding OPNFV: Accelerate NFV Transformation Using OPNFV*. CreateSpace Independent Publishing Platform, 2017.

[35] James Denton. *Learning OpenStack Networking (Neutron)*. Packt Publishing Ltd, 2014.

[36] Mihai Budiu and Chris Dodd. 'The P416 Programming Language.' In: *Operating Systems Review* 51.1 (2017), pp. 5–14.

[37] Jan Kanclirz. *Netcat power tools*. Elsevier, 2008.

[38] *Scapy webpage*. Available online: https://github.com/secdev/scapy (accessed on 04 June 2019). 2019.

[39] Tirumaleswar Reddy et al. *Authenticated and encrypted NSH service chains*. Internet-Draft draft-reddy-sfc-nsh-encrypt-00. Work in Progress. Internet Engineering Task Force, Apr. 2015. 12 pp.