Managing Secure Inter-slice Communication in 5G Network Slice Chains

Luis Suárez(✉), David Espes, Frédéric Cuppens, Cao-Thanh Phan, Philippe Bertin, and Philippe Le Parc

IRT b<>com, Cesson-Sévigné, France
{Luis.Suarez,David.Espes,Frederic.Cuppens,Cao-Thanh.Phan, Philippe.Bertin,Philippe.Le-Parc}@b-com.com

Abstract. Network Slicing is one of the cornerstones for network operators to provide communication services. It is envisioned that in order to provide richer communication services, network slices need to be connected to each other in an orderly fashion, interlacing their functionalities. The challenge is to manage inter-slice communication securely, leveraging on security attributes inherent to the communication service and the constituting network slices.

To solve this inter-slice communication problem, we present a mathematical model based on the concept of Network Slice Chains. This concept helps to specify the end-to-end path of network slices that data must follow for the achievement of the communication service. We propose basic attributes and properties that the Network Slice Chain must comply with in order to be chosen as a valid path for the traffic to flow through. This way, it respects security constraints and assures inter-slice communication obeying the rules stated in the policy.

Keywords: Inter-slice communication · Network Slice Chain · Security · 5G

1 Introduction

Network slicing is one of the key enablers for the use cases that are proposed for 5G [16]. Along with Software Defined Network (SDN), Network Functions Virtualization (NFV) and cloud computing, they provide a novel partitioning scheme to instantiate a Communication Service (CS) on top of network slices. They will use resources that belong to the same Communication Service Provider (CSP) that offers the service or to different operators, organizations and stakeholders [5].

Interactions between network slices will become commonplace, because the CSP can provide common functions through a network slice that is accessible for consumption by other dedicated slices. As network slice interconnection brings

Published by Springer Nature Switzerland AG 2020
A. Singhal and J. Vaidya (Eds.): DBSec 2020, LNCS 12122, pp. 24–41, 2020.
https://doi.org/10.1007/978-3-030-49669-2_2

the risk of exposure to threats from other players, a secure interaction should be guaranteed to minimize security risks. In order to do so, the CSP has to set up different rules and measures to guarantee secure inter-slice communication, knowing beforehand that slices have different security levels, according to their nature and purpose.

An interesting challenge is: how to manage the interactions between network slices when each one has different security attributes and different security requirements? How to bring this to a next level when a chain of slices is considered?

According to our research, as it will be presented in Sect. 2, no work has been made regarding the formal model of a communication service that uses network slices, taking into account their inherent security attributes. Moreover, there is no study about the evaluation of these attributes when inter-slice communication is considered, specially in the case where successive network slices need to be connected. The presented new concepts contribute to go beyond the access control models that already exist (which are more focused on the user or the resources), by adding an end-to-end view of the communication service considering the security needs for its deployment.

Our contribution is three-fold: **(i)** model the network slicing structure mathematically using graph theory, leveraging on the definitions given by Standard Developing Organizations; **(ii)** deduce a general concept called Network Slice Chain, which describes the sequence of network slices that data must flow through in order to provide a Communication Service; and **(iii)** provide properties and policy rules to validate whether the Network Slice Chain can be used, according to the security constraints that are specified in the policy.

The document is organized as follows: Sect. 2 presents works related to inter-slice communication. Section 3 presents an example of a common network slice set-up from a CSP, who will experience challenges regarding the secure composition of a communication service. Section 4 describes the mathematical model, definitions and properties of a Network Slice and Network Slice Chain. Section 5 describes the different components used on the communication model. Section 6 describes the rules and policy validation steps that govern communication in the Network Slice Chain, which are applied to a use case in Sect. 7. After putting into practice these ideas in Sect. 8, Sect. 9 draws concluding remarks.

2 Related Work

Inter-slice access control has attracted few research works. In [4], the 5G-ENSURE project focuses on the access control from end-users to the resources offered by a network slice in a 5G network. They provide a set of countermeasures and enablers for this purpose. The inter-slice communication and access control are not addressed.

In a different perspective, authors in [7] address the inter-slice communication regarding the need to guarantee isolation. They point out that improper inter-slice isolation leads to threats in network slicing. They include the suggestion to use a fine-grained access control to limit access from a tenant to the entire infrastructure.

In [6] the 5G-Monarch project works on providing end-to-end slicing support via enablers pertaining to inter-slice control and management, which are some of the proposed innovations of their work in order to provide slice admission control. The inter-slice management still resides into the Network Slice Management Function (NSMF), as a way to assure that the resources assigned to the network slice instance are optimal, used wisely, at the same time guaranteeing Service Level Agreement (SLA). In the same fashion, authors in [8] propose an inter-slice management mechanism to control events in a 5G network. Using queue and graph theory, they create a reference model that captures events from the network and according to their importance or impact on metrics, classifies the events for resolution, avoiding network congestion. The projects do not provide information about access control mechanisms.

Authors in [14] present how authentication and authorization was integrated in the SONATA Service Platform, in order to manage the authentication, identity management and authorization of users and microservices in a 5G network. Their approach is generic, supplying these security features for the users and the networks functions inside 5G. The slice use case is not mentioned, neither inter-slice communication management.

In [18] authors propose to enhance the Topology and Orchestration Specification for Cloud Applications (TOSCA) modeling language with security parameters. They leverage on the SDN paradigm to use these parameters and, via an access control model, deploy services on Virtual Network Function (VNF) with embedded security countermeasures. In a similar fashion in [11] authors propose an enhancement of the TOSCA language to model the protection of clouds, represented as resources in unikernel system instantiated in virtual machines. Both approaches provide a way to specify and build secured network functions, nonetheless, their approach can be improved by considering a top layer approach from a Communication Service point of view and considering a chain of those VNF of kernels in order to build richer services.

Other authors address the interactions between network functions that are inside a network slice. Leveraging of SDN capabilities, all the required VNF are linked together in a chain in order to manage the traffic as desired. In [10], authors apply this concept to map network slice attributes into the infrastructure of datacenters. Their approach does not cover security attributes or a mathematical modelization. A similar approach is proposed in [20], where a traffic steering solution is implemented for various use cases, keeping in mind a consisted throughput, packet loss, latency and jitter to guarantee quality of service. No slicing consideration or security model is proposed. In [17] authors explore the modelization of network slices considering the allocation of a service instance to a slice instance, according to availability, resources, quality of service and isolation. Even though their model work over the slice instance abstraction, does not consider the interconnection of slices to provide a service.

These works point out challenges, focus on the isolation problem, on how an end user or tenant access to resources, on resource assignment to guarantee a performance rating, and how to perform the inter-slice management and

orchestration via a broker mechanism [5]. No formal model of the network slice environment is provided, neither security considerations for inter-slice communication when several network slices need to be connected to provide composite services. This is a central issue for a CSP that is deploying services via network slices.

3 Motivating Example

In order to better understand the properties and different elements that are inside the proposed model, a use-case scenario is presented. Even though it does not depict a specific service, it is generic enough to fit into any communication service offered by a CSP. The architecture is presented in Fig. 1, which contains a set of Network Slices, connected arbitrarily according to the needs of the CSP. Each Network Slice (NSlice) is configured according to a *service type* to perform a

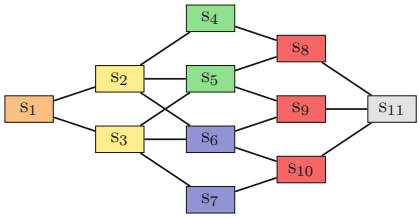


Fig. 1. Topology of the network slices and corresponding types for a CSP. (Color figure online)

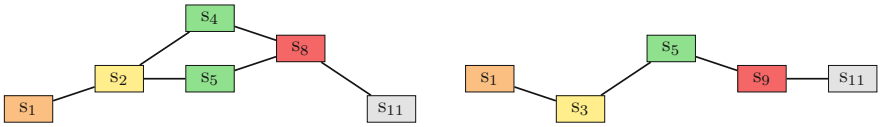
specific function, as specified by 3rd Generation Partnership Project (3GPP) [1]: Enhanced Mobile Broadband (eMBB), Ultra-Reliable and Low Latency Communications (URLLC), and Massive Machine Type Communications (mMTC). These service types are represented by a different color: orange, yellow, green, blue, red and grey.

For example, the orange NSlice could be an aggregation service slice for an enterprise; the yellow slice an IoT slice; green and blue slices constitute added value services (built from network functions to provide services such as traffic filtering, IDS/IPS); the red slice a 5G network slice to provide final connectivity; and the grey slice a data network that provides a concrete service. A more concrete use case illustrating a similar setup is provided in Sect. 7.

All network slices are connected together in an ordered sequence to provide a service. For example, assume the presence of a communication service that we name CS_1. It considers the orange, yellow, green, red and grey service types. Similarly, another communication service named CS_2 has orange, yellow, blue, red and grey service types.

Each Communication Service CS_1 and CS_2 can be set up according to the needs from the CSP by connecting NSlices creating a Communication Service

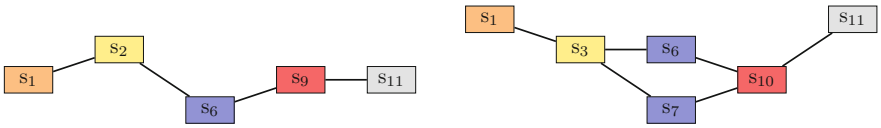
Graph (CSG). For example, regarding CS_1, it can be considered as two CSG: CSG_{11} (Fig. 2a) and CSG_{12} (Fig. 2b). The key message is that, even though the nature of the CS is the same, each slice can have a different configuration and different resources, enabling to provide options of deployment according to the needs. The same approach can be made with CS_2, in which two CSG are



(a) CSG_{11} (b) CSG_{12}

Fig. 2. Communication service graphs for CS_1 and CS_2. (Color figure online)

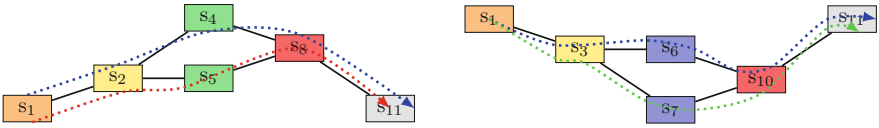
presented: CSG_{21} (Fig. 3a) and CSG_{22} (Fig. 3b). Other arrangements of CSG can be made, enriching the exercise. The advantage of considering the service as a CSG is that the operators can configure the routing of the system in order to forward the traffic through the slices according to a certain policy. With this, traffic can exploit the characteristics of the network topology and then be treated according to the specification of each slice. The traffic will follow a chain of slices that comply with a use-case for the customer. Concretely for CSG_{11},



(a) CSG_{21} (b) CSG_{22}

Fig. 3. Communication service graphs for CS_2

the provider can set up two network slice chains specified by blue and red dotted lines. As presented in Fig. 4a, the blue Network Slice Chain (going through s_1, s_2, s_4, s_8 and s_{11}) covers a green slice with an IDS that detects a certain type of traffic. Similarly, the red Network Slice Chain (going through s_1, s_2, s_5, s_8 and s_{11}) can contain a green slice that has an IDS with a different detection policy. The same approach can be made for CSG_{22}, as shown in Fig. 4b. The presented topology is complex even though the number of network slices is small. As the number of network slices increase, their management becomes a challenge. This manageability has to do with the way to connect the network slices (must ensure the proper authentication and security between them) and how to keep the guarantees of the service offerings to the customer. This implies that the set-up and configuration of the communication service must follow certain rules



(a) CSG_{11} (b) CSG_{22}

Fig. 4. Example of two CSG with two network slice chains. (Color figure online)

and constraints expressed in the policy, which specifies its security requirements and the type of traffic that is allowed to flow. Moreover, as the network is a dynamic entity, the topology can change, so the CSP must perform validation that a path, represented by a Network Slice Chain, can be used for the service required by the customer. Not addressing the needs regarding management and security validation, makes difficult the secure deployment of rich communication services using several connected network slices.

With this setup, the next section elaborates on the properties of Network Slice and Network Slice Chains, whose specification constitutes the major contribution of this work.

4 Network Slice and Network Slice Chain

This Section provides the mathematical background to describe the novel concept called Network Slice Chains. To do so, it evolves from its basic building blocks to then state its key properties.

4.1 Network Service (NS)

The network slicing model relies for its realization on the European Telecommunications Standards Institute (ETSI) NFV concept of Network Service (NS), detailed in [12]. A NS is a composition of network functions arranged as a set of functions with either unspecified connectivity between them or connectivity specified according to one or more forwarding graphs [13]. It is deduced that key components of a NS are Virtual Network Function (VNF), Virtual Link (VL), and VNF Forwarding Graph (VNFFG). All these elements provide a specific functionality and resource requirements for network slices, which will be presented in the next subsection.

4.2 Network Slice (NSlice)

3GPP [2] defines that a CS is offered by a set of Network Slices, being each NSlice composed by an ordered set of NS. This notion of "interconnection" leads us to represent the NSlice as a connected graph.

Definition 1. *The NSlice is a graph composed of:* ***(i)*** *a non-empty set of vertices (V), which are the NS; and* ***(ii)*** *a set of edges (E), which are the VL. For a Network Slice $\mathcal{A}$: $NSlice_{\mathcal{A}}$ = (NS($\mathcal{A}$), VL($\mathcal{A}$)).* □

Property 1. Let $\mathcal{S}$ be the set of NSlices that belong to a CSP. $\mathcal{S}$ = { s_1, s_2, ..., s_i, ..., s_m}. The CSP uses s_i to provide a service to its customers and the disposition of the NSlices obeys the CSP's internal rules and policies. □

Property 2. Each Network Slice has one *service type* that describes its function. The set of types is called $\mathcal{T}_{\mathcal{S}}$. $|\mathcal{T}_{\mathcal{S}}|$ represents the number of *service types* that are provided by the CSP as defined by 3GPP [1]. □

Property 3. The function **type** is used to know the *service type* that the NSlice has. A NSlice can have only one type. The function **type** is defined as **type**: $\mathcal{S} \rightarrow \mathcal{T}_{\mathcal{S}}$. □

Going back to the example in Sect. 3, Fig. 1 helps to represent a set $\mathcal{S}$ of NSlices with its types, identified by a different color.

The CSP uses several interconnected Network Slices to provide a complete service to the customer: this constitutes what is called a Communication Service. Next subsection defines analytically this concept and the inference of the Communication Service Graph.

4.3 Communication Service Graph (CSG)

A CS is defined as an ordered set of types of Network Slices, whose services are offered to different market segments, obeying a business purpose [3]. These Network Slices are connected via Network Slice Links (NSL).

Definition 2. *A type of CS is defined as $\mathcal{T}_{\mathcal{CS}} = \langle \mathcal{T}_{\mathcal{CS}_1}, \mathcal{T}_{\mathcal{CS}_2}, ..., \mathcal{T}_{\mathcal{CS}_m} \rangle$, i.e., the traffic of a CS is going to flow through an ordered set of NSlices. $\mathcal{T}_{\mathcal{CS}} = \langle \mathcal{T}_{\mathcal{CS}_1}, \mathcal{T}_{\mathcal{CS}_2}, ..., \mathcal{T}_{\mathcal{CS}_k} \rangle \mid \forall i \in [1;k], \mathcal{T}_{\mathcal{CS}_i} \in \mathcal{T}_{\mathcal{S}}$.* □

There can exist several NSlices deployed by a CSP for a type $\mathcal{T}_{\mathcal{CS}_i}$. In fact, there exist a set $\mathcal{S}_{\mathcal{T}_{\mathcal{CS}_i}} = \{s \mid \textbf{type(s)} = \mathcal{T}_{\mathcal{CS}_i}\} \in \mathcal{S}$. The interconnection of successive $\mathcal{S}_{\mathcal{T}_{\mathcal{CS}_i}}$, $\mathcal{S}_{\mathcal{T}_{\mathcal{CS}_{i+1}}}$ creates an ordered graph.

Definition 3. *The CSG $\mathcal{C}$ is a directed weighted graph such as: $\mathcal{C}$ = ($\mathcal{S}'$, NSL) where: $\mathcal{S}' = \{s | s \in \mathcal{S} \wedge$ **type(s)** $\in \mathcal{T}_{\mathcal{CS}}\}$ and NSL = $\{(u,v) | u,v \in \mathcal{S} \wedge u \neq v\}$.* □

Property 4. Each link $(u,v) \in$ NSL has a set of attributes $\{a_1, a_2, ..., a_m\}$. (u,v) inherits a quality from graph theory called weight $\mathcal{W}_{(u,v)}$ that is a function which, using the values of the attributes, computes an unified metric for (u,v). $\mathcal{W}_{(u,v)}$ = $\mathcal{F}(a_1, a_2, ..., a_m)$. The definition of $\mathcal{F}$ and the presentation of the attributes are explained in Sect. 5.1. □

These aforementioned definitions and properties help to define a CSG, which provides a way to deploy a concrete communication service and permit the flow of data among a subset to those Network Slices. That is where the concept of Network Slice Chain comes to play, as is shown in the next subsection.

4.4 Network Slice Chain

The Network Slice Chain (NSliceCh) is conceived as a concrete path in the CSG that a flow of data follows, which complies with certain requirements related to the Communication Service purpose, the nature of the traffic and security attributes. The NSliceCh leverages on Definition 3, which defines the CSG as a set of NSlices whose type respects $\mathcal{T}_{\mathcal{CS}}$ over which the traffic will flow. For readability of the definition, $\mathcal{P}$ represents a NSliceCh.

Definition 4. *The CSG $\mathcal{C}$ = ($\mathcal{S}'$, NSL) contains a set of Network Slice Chains $\mathcal{P}_\mathcal{C}$, which comply with the sequence of types of Network slices $\mathcal{T}_{CS}$ and do not form a loop.*

$$\mathcal{P}_\mathcal{C} = \{\langle \mathcal{S}_{\mathcal{P}_1}, ...\mathcal{S}_{\mathcal{P}_i}, ...\mathcal{S}_{\mathcal{P}_m}\rangle \mid \forall\ i \in [1,m], \mathcal{S}_{\mathcal{P}_i} \in \mathcal{S}' \wedge \\ \textbf{\textit{type}}(\mathcal{S}_{\mathcal{P}_i}) = \mathcal{T}_{\mathcal{CS}_i} \wedge \nexists\ \mathcal{S}_{\mathcal{P}_i} \in \langle \mathcal{S}_{\mathcal{P}_{i+1}}, ..., \mathcal{S}_{\mathcal{P}_m}\rangle\}$$

□

A *security constraint* refers to the factors that impose restrictions and limitations on the system or actual limitations associated with the use of the system [19]. Applied to the subject under consideration, a security constraint refers to the requirements that a system should comply with in relation to security parameters. Examples could be the encryption level of a Virtual Private Network, or the protocol that must be used in a communication. These requirements are stated in the policy, which, as a system, makes sure it is enforced as needed.

In Fig. 4a two different NSliceCh are shown: one in red and the other in blue dotted line. It is supposed that they comply with the demands from the CS and its security constraints.

With all the previous definitions, all the elements are provided in order to use the tools to assess inter-slice communication.

5 Operators and Elements Involved in Inter-slice Communication

From the mathematical representations, definitions and properties shown in Sect. 4, we define operators and elements needed to manage inter-slice communication. These are the *attributes* of Network Slices and their corresponding measurement using *metrics*.

5.1 Attributes

Attributes refer to a feature or property of an entity [15]. Since entities are diverse in nature and functionality, it is difficult to have a complete list of attributes, specially for security requirements. For this proposal, attributes that are considered important from a security perspective are Affinity (Af), Trust (T) and

Security Level (SL). In this subsection, operations are proposed among them, being these operations a particular case of the function $\mathcal{F}$ stated in Property 4.

Let $\mathcal{C} = (\ \mathcal{S}'$, NSL). Each NSlice $s \in \mathcal{S}'$ has a set of attributes defined as $\mathcal{A}(s) = \{(a_i, v_i) \mid a_i \in \{\text{Af, T, SL}\}, v_i \in \mathbb{R} \wedge \forall j \in [1, |\mathcal{A}|] \backslash \{i\}\ a_j \neq a_i\}$.

Each attribute is defined and specified according to formulas and properties as follows:

Affinity (Af): It is used to avoid conflicts regarding the nature of the offered slices, helping to determine whether they can be connected or can coexist.

Affinity has a nominal type of data, specified by the network administrator. Considered values are the basic *service types* for 5G established by 3GPP with the addition of a *common service type* that contains regular functionality and aids to connect dissimilar NSlices.

Property 5. Affinity for a link $(s_i, s_j) \in$ NSL is achieved if the (s_i, s_j) that make it up have the same affinity parameter. We call $\mathcal{F}_{\text{Af}}$ the function that finds the affinity for a link $(s_i, s_j) \in$ NSL.

$$\mathcal{F}_{\text{Af}} : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}$$

$$\forall s_i, s_j \in \mathcal{S}' \text{ NSL }, \exists\ (a_{i_p}, v_{i_p}) \in \mathcal{A}(s_i) \wedge (a_{j_k}, v_{j_k}) \in \mathcal{A}(s_j) \mid a_{i_p} = a_{j_k} = \text{Af} \Rightarrow$$

$$\mathcal{F}_{\text{Af}}(s_i, s_j) = \begin{cases} 1, \text{if } v_{ip} = v_{jk} \\ 0, \text{otherwise} \end{cases}$$

This means that if the services belong to the same *service type*, their affinities are the same and the function will have 1 as a result. □

Property 6. Affinity for a NSliceCh $\mathcal{P}$:

Let $\mathcal{C} = (\ \mathcal{S}'$, NSL). $\forall\ \mathcal{P} = \langle s_1, s_2, ..., s_n \rangle \in \mathcal{P}_C \wedge \forall\ s_i \in \mathcal{S}' \wedge (s_i, s_{i+1}) \in$ NSL:

$\mathcal{G}_{\text{Af}} : \mathcal{P}_C \rightarrow \mathbb{R}$ with: $\mathcal{G}_{\text{Af}}(\mathcal{P}) = \prod_{i=1}^{n-1} \mathcal{F}_{\text{Af}}(s_i, s_{i+1})$

This means that for a chain of network slices, the result for affinity is the product of values of this attribute for each of the links that belongs to the NSliceCh. □

Corollary 1. *Affinity for a NSliceCh is achieved as a consequence of Property 5, since the NSliceCh is a subset of the CSG.*

Trust (T): It denotes the confidence to establish a business relation, enabled by the acknowledgement of the identity of the other party. Trust has an ordinal type of data, enabling to have levels of trust, for example, {trusted, not-trusted}, or equivalently, {1, 0}.

Property 7. Intuitively, the trust level of the destination NSlice has to be at least greater or equal to the trust level of the source NSlice.

We call $\mathcal{F}_{\mathrm{T}}$ the function that finds the trust for a link $s_i, s_j \in$ NSL.

$$\mathcal{F}_{\mathrm{T}} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$$
$$\forall s_i, s_j \in \mathcal{S}', \exists\ (a_{i_p}, v_{i_p}) \in \mathcal{A}(s_i) \wedge (a_{j_k}, v_{j_k}) \in \mathcal{A}(s_j) \mid a_{i_p} = a_{j_k} = \mathrm{T} \Rightarrow$$
$$\mathcal{F}_{\mathrm{T}}(s_i, s_j) = \begin{cases} 1, \text{if } v_{ip} \geq v_{jk} \\ 0, \text{otherwise} \end{cases}$$

This means that if the trust of the links are at least the same, the function will have 1 as a result. □

Property 8. Trust level for a NSliceCh $\mathcal{P}$:

Let $\mathcal{C} = (\ \mathcal{S}'$, NSL). $\forall\ \mathcal{P} = \langle s_1, s_2, ..., s_n \rangle \in \mathcal{P}_C \wedge \forall\ s_i \in \mathcal{S}' \wedge (s_i, s_{i+1}) \in$ NSL:

$\mathcal{G}_{\mathrm{T}} : \mathcal{P}_C \to \mathbb{R}$ with: $\mathcal{G}_{\mathrm{T}}(\mathcal{P}) = \prod_{i=1}^{n-1} \mathcal{F}_{\mathrm{T}}(s_i, s_{i+1})$

This means that for a chain of network slices, the result for trust is the product of values of this attribute for each of the links that belongs to the NSliceCh. □

Corollary 2. *The trust in a NSliceCh is obtained as an extension of the trust value in the links which embed it.*

Security Level (SL). It shows the rating of the slice in terms of security, for example its confidentiality, integrity or other criteria that can be measured for the slice internal components. SL has an ordinal type of data, making possible to create, as its name implies, security levels to classify NSlices and manage their communication. The quantity of levels depends on the use case and need, as well as the criteria used to find its rating. For example, {high, medium, low}, or equivalently {3, 2, 1}.

Property 9. The intuition is that the SL of the destination NSlice has to be at least as high as the SL of origin NSlice:

We call $\mathcal{F}_{\mathrm{SL}}$ the function that finds the Security Level for a link $s_i, s_j \in$ NSL.

$$\mathcal{F}_{\mathrm{SL}} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$$
$$\forall s_i, s_j \in \mathcal{S}', \exists\ (a_{i_p}, v_{i_p}) \in \mathcal{A}(s_i) \wedge (a_{j_k}, v_{j_k}) \in \mathcal{A}(s_j) \mid a_{i_p} = a_{j_k} = \mathrm{SL} \Rightarrow$$
$$\mathcal{F}_{\mathrm{SL}}(s_i, s_j) = min(s_i, s_j)$$

The outcome of this function is the minimum value of SL for the considered links. □

Property 10. Security Level for a NSliceCh $\mathcal{P}$:

Let $\mathcal{C} = (\ \mathcal{S}'$, NSL). $\forall\ \mathcal{P} = \langle s_1, s_2, ..., s_n \rangle \in \mathcal{P}_C \wedge \forall\ s_i \in \mathcal{S}' \wedge (s_i, s_{i+1}) \in$ NSL:

$\mathcal{G}_{\mathrm{SL}} : \mathcal{P}_C \to \mathbb{R}$ with: $\mathcal{G}_{\mathrm{SL}}(\mathcal{P}) = \min_{i=1}^{n-1} \mathcal{F}_{\mathrm{SL}}(s_i, s_{i+1})$

This means that for the NSliceCh the minimum value is used as a way to portray the lowest security level admitted on the path. □

5.2 Metrics

According to [21], a metric is a standard of measurement that describes the conditions and the rules for performing a measurement of a property and for understanding the results of a measurement. A metric provides knowledge about an entity via its properties and the measured values obtained for that property. In our case, metrics are associated to links. For every link $(s_i, s_j) \in \text{NSL}$, it exists a metric vector m.

It is defined as: $m_{(s_i,s_j)} = \{(\text{Af}, \mathcal{F}_{\text{Af}_{(s_i,s_j)}}), (\text{T}, \mathcal{F}_{\text{T}_{(s_i,s_j)}}), (\text{SL}, \mathcal{F}_{\text{SL}_{(s_i,s_j)}})\}$.

5.3 Final Remarks

After stating the attributes for Network Slices, the metrics and the functions to perform operations on them, the set of tools needed to validate a Network Slice Chain is complete. This compliance with a security policy is presented in the next Section.

6 Policy Validation for Network Slice Chains

The inter-slice communication depends on the use case and the service type of the NSliceCh. Somehow, the communication should be regulated according to certain *rules* r_i that are grouped in a policy Π. Specifically, rules are expressed as a vector $\langle$ Subject $\mathcal{SU}$, Object $\mathcal{O}$, Security Constraint SC, Permission $\rangle$ and its components specify the conditions for communication. This Section presents these components along with a *compliance operator* and the mechanisms to *validate the compliance* with the policy.

6.1 Entities: Subjects and Objects

Entities indicate the name of the actors that interact in the topology. Specifically, the entity called **subject**, denotes the active entity, refers to the NSlice that requests a service. The passive entity, the **object**, refers to the NSlice that receives the request. Subjects $\mathcal{SU}$ and objects $\mathcal{O}$ are represented as sets:

$\mathcal{SU} = \{su_1, su_2, ..., su_n\}$; $\mathcal{O} = \{o_1, o_2, ..., o_n\}$.

6.2 Security Constraint

Security Constraint, denoted by SC, represents the security conditions that the path has to comply with i.e., each link of the path must guarantee a security attribute superior or equal to the one specified in the rule. It is defined as follows:

$$\text{SC} = \{(a_i, v_{i_{min}}) \mid a_i \in \{\text{Af, T, SL}\},\ v_{i_{min}} \in \mathbb{R} \wedge \forall j \in [1, |\mathcal{A}|] \backslash \{i\}\ a_j \neq a_i.$$

6.3 Permission

Describes the ability to perform an operation on a protected object or resource. Considered actions for can be to *allow* or *deny* the operation after its evaluation.

6.4 Compliance Operator

Denoted by $\cong$, its purpose is to validate if the metrics of a link $(s_i, s_j) \in$ NSL complies with the security constraints SC_i of the rule r_i. It is defined as:

$$\forall (a_k, v_k) \in m_{(s_i,s_j)}, \exists\ (a_p, v_p) \in \mathrm{SC}_i \mid a_k = a_p \wedge v_k \geq v_p) \Leftrightarrow m_{(s_i,s_j)} \cong \mathrm{SC}_i$$

This means that for each set of attributes specified for a subject, it needs to exist a pair of the same name of attributes for the object. Subject and object refer to Network Slices, that is, the link between them that complies with the security constraint.

The $\geq$ symbol, the *greater or equal to* operator, provides a way to compare quantitatively the values of the attributes. It specifies the preference to communicate with an object that has a security attribute having a greater or equal value. This is clarified better with an example in Sect. 7.

6.5 Rule for Policy Validation

It is necessary to verify that at least one NSliceCh, represented by $\mathcal{P}$, exists and complies with the metric in the policy.

SC corresponds to the constraints that must be respected, that is to say, that a path $\mathcal{P} \in \mathcal{P}_C$ in a CSG C matches the criteria if: $\mathcal{G}_{\mathrm{Af}}(\mathcal{P}) \geq \mathrm{SC}_{\mathrm{Af}} \wedge \mathcal{G}_{\mathrm{T}}(\mathcal{P}) \geq \mathrm{SC}_{\mathrm{T}} \wedge \mathcal{G}_{\mathrm{SL}}(\mathcal{P}) \geq \mathrm{SC}_{\mathrm{SL}}$. This means that not only the evaluation of each one of the attributes should be greater than the ones specified by the constraints in SC, but also that all those evaluations should agree.

Property 11. A path $\mathcal{P} = \langle s_1, s_2, ..., s_n \rangle$ complies with a rule r_i if each link of $\mathcal{P}$ complies with the security constraints SC_i of r_i such as:

$$\mathcal{P} \cong \mathrm{SC}_i \Leftrightarrow \forall\ j \in [1, n-1],\ m_{(s_j,s_{j+1})} \cong \mathrm{SC}_i$$

□

Property 12. A CSG $\mathcal{C}$ complies with the policy if at least a path exists that fulfils the constraints for each rule of the policy. □

Property 13. For an end-to-end NSliceCh, trust and affinity compliance are enforced if the product of all the computed trust values of the NSL that constitute the NSliceCh has a result of 1. This can be inferred from Property 7 and Property 5 respectively.

Property 14. For an end-to-end NSliceCh, security level compliance is achieved if the SL values of the NSL that conform the NSliceCh are superior to the minimum value established by the policy.

Corollary 3. *The compliance with the SL for a NSliceCh is guaranteed since the NSliceCh is a subgraph of the CSG.*

After this verification of the policy, the CSP can be warned about rules that are not satisfied because the deployed Network Slices do not meet a security criteria. In consequence, the CSP can either add other network slices that meet the security constraints or soften the security policy.

6.6 Discussion

The rules presented in this Section provide assurance that the components of a NSliceCh comply with the constraints expressed in the policy. Moreover, they enforce not only compliance but that the metric meets a certain level stated in the policy. At the same time, special care has to be taken when including a high number of attributes, which can render difficult the task to find a NSliceCh due to the fact that the problem cannot be solved in polynomial time (it is exponential). This approach gives way to think about a more complex scenario, where there could be a possibility that two NSliceCh exist, and the policy helps to choose the best one according to the security requirement. This will be shown via a concrete use case in the following Section.

7 Use Case

This section describes with a use case the way in which the rules stated in Sect. 6 are applied. The topology is shown in Fig. 5, which leverages on the CSG_{11} from Fig. 4a.

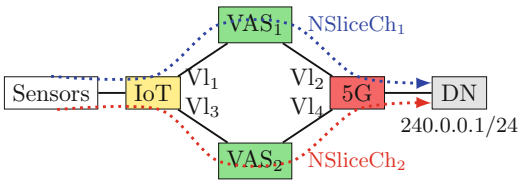


Fig. 5. Topology of a use-case scenario involving inter-slice interactions.

7.1 Description

The CSP has four network slices, one for a 5G network, another used by a customer (a tenant) that is configured for an IoT service, and two intermediate network slices that provide value-added services (VAS), such as analytics, traffic filtering or monitoring.

The sensors operating under the IoT slice use the services provided by the IoT network slice. Nonetheless, there could be some special devices that need to have access to a specific server on the Internet. To this end, the idea is to allow this specific connectivity using the 5G Core (5GC) slice as a bridge to reach the server hosted in a Data Network (DN) in the Internet. The attributes and corresponding metrics for the entities involved in this interaction are specified in Table 1. Similarly, the policy Π states that communication is allowed only if the proposed NSliceCh has a minimum SL of medium (numeric value 2). The objective is to check the validity of coherence of policy: verify the existence of the NSliceCh that complies with the policy, so the communication is authorized.

Table 1. Parameters for the elements in the example scenario

	IoT slice	VAS_1 slice	VAS_2 slice	5G slice
Af	mIoT	Common	Common	eMBB
SL	Medium:2	Low:1	Medium:2	High:3
T	Y	Y	Y	Y

7.2 Validation of Compliance of the NSliceCh

It is assumed that the topology represented in Fig. 5 represents a CSG from a CSP that provides a CS and it is possible to find a NSliceCh as a sequence of NSlices and NSL. Considering the concept of connectivity and directed-graph characteristics of the outbound traffic, there exist two NSliceCh:
$\text{NSliceCh}_1 = \langle \text{IoT}, \text{VAS}_1, \text{5G} \rangle$
$\text{NSliceCh}_2 = \langle \text{IoT}, \text{VAS}_2, \text{5G} \rangle$

Compliance for Affinity. The need is to connect an IoT slice to a 5G Slice (which are dissimilar) via an intermediary slice that has a common functionality. $\mathcal{G}_{\text{Af}}(\text{NSliceCh}_1) = 1$, because IoT-$\text{VAS}_1$ and VAS_1-5G have compliant affinity values. Similarly, $\mathcal{G}_{\text{Af}}(\text{NSliceCh}_2) = 1$, because IoT-$\text{VAS}_2$ and VAS_2-5G have compliant affinity values. Both Network Slice Chains are compliant with this requirement.

Compliance for Trust. From Table 1, it is inferred that the CSP trusts its tenant, its services and they have good business relationship. $\mathcal{G}_{\text{T}}(\text{NSliceCh}_1) = \mathcal{G}_{\text{T}}(\text{NSliceCh}_2) = 1$, because the trust level of the source and destination Network Slices are equal.

Compliance for Security Level. This attribute obeys Property 9, 14 and Corollary 3. $\mathcal{G}_{\text{SL}}(\text{NSliceCh}_1) = 1$, since it is the lowest SL on this path.
$\mathcal{G}_{\text{SL}}(\text{NSliceCh}_2) = 2$, since it is the lowest SL on this path.

NSliceCh_2 complies with the requirement by traversing two consecutive network slices with medium security level to then go into a high security level network slice. From the evaluation of the attributes, it can be concluded that NSliceCh_2 is the one that complies with what is stated in the policy Π.

7.3 Discussion

The example depicts a use case that will become usual inside a CSP network. The CSP can have an orchestrator that automatically chooses the NSliceCh that complies with the policy. If there exists a NSliceCh that respects the constraint, it will be selected. If it is not the case, the CSP will know and can adjust the configuration of the NSlices or the policy to comply with the policy. This enables

a CSP to have tools to compare different NSliceCh according to their security characteristics and choose the best one. This approach is extensible to other type of metrics such as latency, performance or cost.

8 Implementation

A test-bed was set up in order to verify the proposed approach to manage inter-slice communication. It uses:

- TOSCA as a specification to describe service components, their relationships and its orchestration, in order to create a Service Template that can be implemented in diverse cloud environments [9].
- Tacker as Generic VNF Manager (VNFM) and an NFV Orchestrator (NFVO) to deploy and operate Network Services and VNF on a NFV infrastructure platform as Openstack.
- Openstack Heat as orchestration engine to launch multiple composite cloud applications based on templates.
- Openstack Neutron and Nova as orchestrators for network connectivity and compute capabilities towards the infrastructure.

The architecture is shown in Fig. 6, where the process to setup the CSP environment is specified by six steps: **(1)** VNF, NS, VNFFG TOSCA templates are

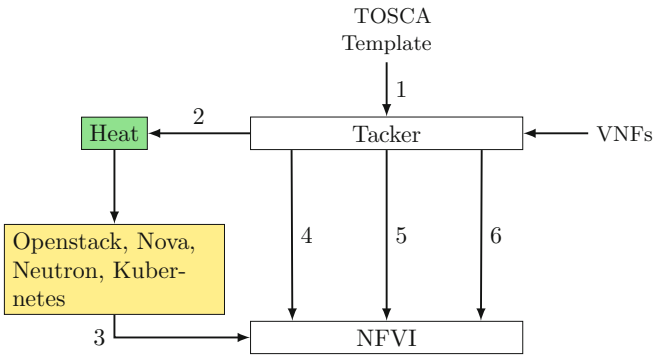


Fig. 6. Topology of a use-case scenario involving inter-slice interactions.

deployed into Tacker; **(2)** Tacker instructs Heat to perform VNF onboarding, orchestration and LCM; **(3)** Openstack Nova and Neutron triggers deployment into the NFVI; **(4)** the service is configured; **(5)** the VNFFG is installed on Open vSwitch (OVS) via an SDN Controller; and **(6)** deploy the set-up of the monitoring scheme for the service.

The architecture shown in Fig. 6 is used to deploy CSG consisting of 20, 40, 60 and 100 network slices. Each one has connections with a security level

constraint (low, medium or high). A single policy is implemented, which dictates that connectivity between slices is allowed only if the security level of the next network slice is equal or higher than the origin network slice.

The algorithm proceeds to scan all possible paths from source to destination slice, and evaluates whether the found paths comply with the policy. Figure 7 shows with a blue line the results for the experiment, conceived to tell the time spent to find the first valid path that complies with the policy. For instance, for a CSG composed of 100 network slices, 1318 valid paths were found and only 0.22 ms were needed to find the first valid path.

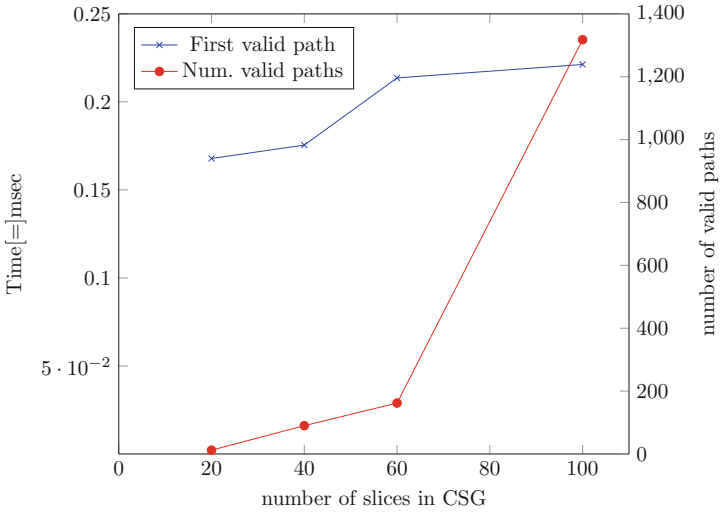


Fig. 7. Time to obtain the first valid path according to policy (in blue) and the number of valid paths (in red) for a CSG of 20, 40, 60 and 100 network slices (Color figure online)

From Fig. 7 it can also be inferred that as the number of network slices increases, the time that is needed to calculate one valid path that satisfies the policy increases as well. This is related with the increment in the number of valid paths that are found, shown with a red line in the same Figure. Nonetheless, a duration of up to 0.22 ms for the case with 100 network slices is valuable, because it is low enough to be used in real-time to find a path that satisfies a policy.

9 Conclusions

The utilisation of network slices as a mechanism to provide communication services to customers and tenants will become commonplace, as technology becomes mature and adoption of enabling technologies such as NFV and SDN increases. Since the nature of a network slice is conceived as a unit specially assembled

for a certain use case, the creation of rich end-to-end communication services necessarily involves the communication between several network slices. From a security perspective, the interconnection of the slices must obey policies that guarantee secure interactions and enable just the required traffic between them. In this paper the concept of Network Slice Chain is defined, leveraging from the definitions provided by ETSI and 3GPP and empowered by graph theory.

These elements are used in the communication model that **(1)** assures that there is a Network Slice Chain connecting the required network slices; **(2)** that the Network Slice Chain complies with the constraints expressed in the policy; and **(3)** assures that beyond compliance, it has a minimum rating level compared to what is needed in the policy. These three elements provide a secure environment for the CSP and its tenants.

Regarding the experimental results, the execution time to find a compliant path for a CSG is low, which permits to use our proposition to compute the validation of a security policy in real-time. The objectives for the future will be to find the best path among all the paths that respect the security policy.

The proposed inter-slice communication model is extensible for application in any service and for the inclusion of other security attributes, so security requirements can be expressed more richly. It complies with any access control model, ensuring a straightforward implementation.

References

1. 3GPP: Specification # 23.501 (2018)
2. 3GPP: Specification # 28.531 (2018)
3. 3GPP: Specification # 28.801 (2018)
4. 5G-ENSURE: Deliverable D2.7 - Security Architecture (2016)
5. 5G-PPP: View on 5G Architecture (Version 2.0) (2017)
6. 5G-PPP: D2.3, 5G Mobile Network Architecture, Final overall architecture (2019)
7. Americas, G.: The Evolution of Security in 5G (2019)
8. Bordel, B., Alcarria, R., Sánchez-de-Rivera, D., Sánchez, Á.: An inter-slice management solution for future virtualization-based 5G systems. In: Barolli, L., Takizawa, M., Xhafa, F., Enokido, T. (eds.) AINA 2019. AISC, vol. 926, pp. 1059–1070. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-15032-7_89
9. Brogi, A.: TOSCA in a nutshell: promises and perspectives (2014)
10. Clayman, S., Tusa, F., Galis, A.: Extending slices into data centers: the VIM on-demand model. In: 2018 9th International Conference on the Network of the Future (NOF), pp. 31–38 (2018)
11. Compastié, M., Badonnel, R., Festor, O., He, R.: A TOSCA-oriented software-defined security approach for unikernel-based protected clouds. In: 2019 IEEE Conference on Network Softwarization (NetSoft) (2019)
12. ETSI: ETSI GR NFV-EVE 012 V3.1.1 (2017–12) (2017)
13. ETSI: ETSI GS NFV-IFA 014 V2.3.1 (2017–08) (2017)
14. Guija, D., Siddiqui, M.S.: Identity and access control for micro-services based 5G NFV platforms. In: Proceedings of the 13th International Conference on Availability, Reliability and Security. ACM (2018)

15. Herrmann, D.: Complete Guide to Security and Privacy Metrics: Measuring Regulatory Compliance, Operational Resilience, and ROI. CRC Press, Boca Raton (2007)
16. Marsch, P., Bulakci, O., Queseth, O., Boldi, M.: 5G System Design: Architectural and Functional Considerations and Long Term Research, 1st edn. Wiley, Hoboken (2018)
17. Nowak, T.: Matematyczny model izolacji usług w sieciach plastrowych. Przeglkad Telekomunikacyjny + Wiadomości Telekomunikacyjne (2017)
18. Pattaranantakul, M., He, R., Zhang, Z., Meddahi, A., Wang, P.: Leveraging network functions virtualization orchestrators to achieve software-defined access control in the clouds. IEEE Trans. Dependable Secure Comput., 1 (2018)
19. Ross, R., McEvilley, M., Oren, J.: Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems. Technical report, NIST Special Publication (SP) 800–160, vol. 1, National Institute of Standards and Technology (2018)
20. Trajkovska, I., et al.: SDN-based service function chaining mechanism and service prototype implementation in NFV scenario. Comput. Stand. Interfaces **54**, 247–265 (2017)
21. de Vaulx, F.J., Simmon, E.D., Bohn, R.B.: Cloud Computing Service Metrics Description. Special Publication (NIST SP) - 500–307 (2018)