A Novel Network Slicing based Security-as-a-Service (SECaaS) Framework for Private 5G Networks

Shalitha Wijethilaka*, Madhusanka Liyanage†
*†School of Computer Science, University College Dublin, Ireland, †CWC, University of Oulu, Finland
Email: *mahadurage.wijethilaka@ucdconnect.ie, †madhusanka@ucd.ie, †madhusanka.liyanage@oulu.fi

***Abstract*—Fifth Generation (5G) and beyond telecommunication networks introduce a set of novel technologies such as private 5G networks, also known as Local 5G Operator (L5GO), and Network Slicing (NS) for the realization of diverse novel applications with heterogeneous network requirements. Among other network-level requirements, security is a critical challenge in L5GO networks. Facilitating the security requirements in an NS-enabled L5GO environment while increasing resource utilization efficiency is arduous. This paper introduces a novel security framework for NS-enabled L5GO networks. The proposed framework increases the scalability, dynamicity, and flexibility of the system while focusing on reducing the cost. Moreover, this paper verifies the functionality of the security framework using a real testbed. Extensive experiments are performed to analyze the framework's behaviour in terms of resource conservation, cost reduction, and latency variation.**

***Index Terms*—Security, Security as a Service, Network Slicing, L5GO, 5G, Private 5G Networks, OpenStack, OpenBaton**

I. Introduction

Private 5G network, also known as Local 5G Operator (L5GO), is a novel advent in 5G and beyond networks which supposed to facilitate network requirements of localized environments such as hospitals, schools, universities, and industrial environments. This novel paradigm allows users to provision their own 5G ecosystems with a unique design to provide operation-specific network requirements [1]. Network slicing (NS) is another key technology in future networks that allows dividing the physical network into multiple logical networks, known as network slices, to provide specific network requirements for different applications. In [2], Siriwardhana et al. identified NS as a significant technology in private 5G networks. Multiple verticals and their tenants with diverse network requirements of private 5G networks can be efficiently served with NS [3].

Security is a significant challenge in any telecommunication network. In private 5G networks, several security mechanisms are also required to be implemented to secure the network environment. However, as private 5G networks are implemented to facilitate network requirements of a particular environment, security challenges that can be transpired in network slices in these networks are nearly identical. Thus, almost the same set of security functions needs to be deployed in all the network slices. Pertinently, private 5G networks have a limited network resource amount. Therefore, the available network resource amount should be carefully managed within the network. Moreover, the security attack space is continuously evolving. Also, the severity of a particular security attack varies with time. Hence, security mechanisms should be updated dynamically. This paradigm surfaces the requirement of a novel concept called Security as a Service (SECaaS) [4].

Therefore, this paper proposes the novel concept of an NS-based SECaaS solution for mitigating security attacks in L5GO environments. We introduce the possibility of deploying a dedicated security slice and a Security Function repository (SFR) that manages the Security Functions (SFs) in the NS ecosystem to provide security services. NS-based security frameworks have been presented in [5]–[7]. In [3], [8], authors implemented NS frameworks for private 5G networks. However, the existing literature lacks researches on NS-based security frameworks specific to L5GO environment. To the best of our knowledge, this is the first research in this area to present an NS-based SECaaS solution to private 5G environments while optimizing the network resource utilization. We develop the high-level architecture of the proposed solution. The proposed framework presents two modes of operation, i.e., default and fallback, and two deployment options, i.e., static and dynamic. An extensive feasibility analysis is performed, and the solution is verified using a real testbed. Also, the potential challenges are discussed in the paper for the completeness of the solution.

The rest of the paper is organized as follows. Section II presents the architecture of the proposed solution. Feasibility evaluation, performed to evaluate the functionality and the performance of the framework using real experiments and simulations, is presented in III. Section IV compares the framework with existing works, and discusses the implementation challenges. Finally, section V concludes the paper.

II. Proposed SecaaS NS Architecture

Discussed challenges in private 5G networks entail a novel security solution. Our proposed architecture for SECaaS via NS is shown in figure 1. It mainly consists of two parts, i.e., Dedicated Security slice and SFR. The security slice is an End-to-End (E2E) network slice that is designed to deploy SFs to provide security services to other slices. SFR manages the available SFs. The primary objective of the solution is to provide the security services dynamically and manage the available network resources for security services efficiently. This allows better security attack mitigation in an L5GO environment.

While creating the dedicated security slice, a sufficient amount of resources to cater the required security services is allocated to the security slice beforehand. When there is a

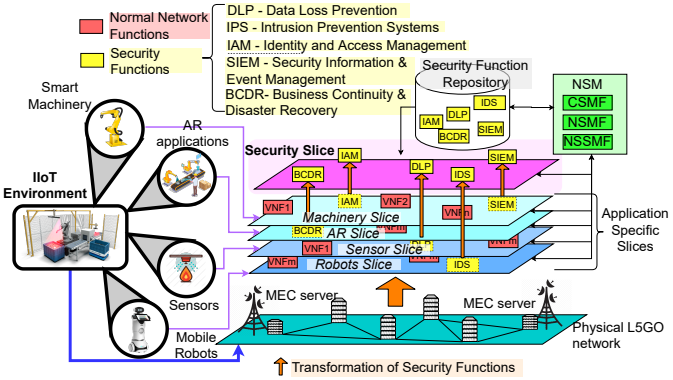

Fig. 1: Proposed security framework

TABLE I: Notations

Symbol	Definition
N	Number of network slices in the system
K	Number of SFs in a slice
B	Number of basic SFs in a normal slice
M	Number of SFs in security slice
$R_{X=k}$	Resources in k^th slice
$R_{X=k,F=l,D}$	Resources for deployment of l^{th} SF in k^{th} slice
$R_{X=k,F=l,P}$	Resources for processing of l^{th} SF in k^{th} slice
$R_{X=S}$	Resources in security slice
R_{T_T}	Total resources for security in traditional system
R_{T_P}	Total resources for security in new system
R_A	Saved resources from new framework
$R_{A\|FM}$	Saved resources from new system in fallback mode
$R_{T_{P\|FM}}$	Resources for security in new system in fallback mode
P_i	Allocated resource percentage from i^th slice
$R_{X=S\|UT}$	Upper threshold of resources in security slice
$R_{X=S\|LT}$	Lower threshold of resources in security slice
$R_{X=k\|U}$	Resource utilization in k^{th} slice
$R_{X=k\|A}$	Resource allocation to k^{th} slice

specific security requirement from an L5GO user, the corresponding SF will be selected from the SFR and dynamically migrated to the security slice. The packet flow of a particular L5GO user needs to be diverted from the relevant slice to the security slice to acquire the security service.

The L5GO can dynamically direct their security requirements to the Network Slice Manager (NSM) via Security Service Level Agreements (SSLAs). The NSM forwards the security requirements to the SFR. Then, it identifies the required SFs to facilitate security requirements and migrates identified SFs to the security slice. New security solutions can also be registered in the SFR. Hence, the SFR continuously evolves to mitigate diversified security attacks.

The resource variation in the system can be formulated as follows. The symbols used in the paper are shown in table I.

$$R_{T_N} = \sum_{i=1}^{N}\sum_{j=1}^{K}(R_{X=i;F=j;D} + R_{X=i;F=j;P}) \quad (1)$$

As deployment overhead is the same for all slices (assumption: since the same attack space is affected to all the slices in the L5GO environment, the same set of security functions need to be deployed in all slices)

$$\sum_{j=1}^{K} R_{X=1;F=j;D} = \sum_{j=1}^{K} R_{X=2;F=j;D} = \ldots = \sum_{j=1}^{K} R_{X=N;F=j;D} \quad (2)$$

Therefore,

$$\sum_{i=1}^{N}\sum_{j=1}^{K} R_{X=i;F=j;D} = N \sum_{j=1}^{K} R_{F=j;D} \quad (3)$$

$$R_{T_T} = N \sum_{j=1}^{K} R_{F=j;D} + \sum_{i=1}^{N}\sum_{j=1}^{K} R_{X=i;F=j;P} \quad (4)$$

$$R_{T_N} = \sum_{j=1}^{K} R_{F=j;D} + \sum_{i=1}^{N}\sum_{j=1}^{K} R_{X=i;F=j;P} \quad (5)$$

$$R_A = R_{T_T} \quad R_{T_N} \quad (6)$$

$$= (N \quad 1) \sum_{j=1}^{K} R_{F=j;D} \quad (7)$$

Equation 7 shows the amount of resources that can be saved from our security framework. This amount of resources can be allocated to alleviate the traffic fluctuations in the system.

A. Modes of operations

The proposed framework can be operated under two basic modes of operations: Default mode and Fallback mode. Each mode has its own advantages and disadvantages.

1) Default Mode

In this mode, all the SFs in the NS ecosystem are operated via the security slice, i.e., no SFs are implemented in each slice. When a specific slice requires a security service, it is mandatory to get it via the security slice. The complete traffic flow of the tenant slice needs to divert to the security slice. Added latency is higher in this mode due to inter-slice communication to get each security requirement. However, in terms of resource utilization, this mode performs well. The equation 5 shows the amount of resources in the security slice under this mode of operation.

2) Fallback Mode

Here, the proposed framework acts as the fallback option for the slicing ecosystem for security operations. A basic set of SFs is deployed in tenant slices. If the deployed SFs can not facilitate a specific security requirement, it receives the security service via the security slice. Diverting traffic between slices is at a minimal level in this mode. However, when considering the resource utilization efficiency, it is at a lower level due to the deployment of the same SFs in multiple slices. Resource utilization in this mode can be formulated as follows.

$$M = K \quad B \quad (8)$$

$$R_{AjFM} = R_{T_T} \quad R_{T_{P|FM}} \quad (9)$$

$$R_{AjFM} = N \quad (\sum_{i=1}^{K}(R_{F=i;D}) \quad \sum_{i=1}^{B}(R_{F=i;D})) \quad \sum_{i=1}^{M}(R_{X=S;F=i;D}) \quad (10)$$

B. Life Cycle Management (LCM) of the Security Architecture

The NSM plays the management role of the proposed security architecture. It manages the LCM of the security slice. The NSM performs SF deployment in the security slice according to the security requirements through the SFR. According to the resource requirements, the allocated network resources for the security slice need to be managed among deployed SFs. Furthermore, the NSM handles the diversion of the traffic from a tenant slice to the security slice and backwards. Hence, the traditional NSM needs to be upgraded to execute these functionalities.

C. Deployment methodologies of the security slice

Primitively, we have two options when deploying the proposed framework in an L5GO environment: static deployment and dynamic deployment.

1) Static Deployment

In this method, constant network resources from each slice are allocated to create the security slice. As the security slice continuously owns a sufficient amount of resources, it can execute security operations faster. However, unutilized network resources can be found when there are no security-threatening incidents. The algorithm 1 can be used to create the security slice with the required resources.

Algorithm 1 An algorithm for static resource sharing

R_{ST} 0 ▷ Total resources for security slice
R_{SS} 0 ▷ Total resources required for SFs
while s in Slices **do**
  R_{ST} $R_{X=s}$ P_s + R_{ST}
end while
while sf in SecurityFunctions **do**
  R_{SS} $R_{F=sf}$ + R_{SS}
end while
if $R_{ST} < R_{SS}$ **then**
  for k = 1, k++, while $k < K$
    Recalculate P_k
end if

2) Dynamic Deployment

The allocated amount of network resources dynamically varies under dynamic deployment method. This method increases resource utilization efficiency by allocating resources as required depending on the severity of the security incident. Therefore, resources can be utilized for other operations when there are no security incidents. However, additional computing and processing overheads need to be allocated to grant resources between slices dynamically. The same algorithm in 1 can be used to create the security slice initially in this method. For dynamic resource allocation in the system, algorithm 2 can be used. The NSM should execute this algorithm periodically while collecting the information related to resource utilization of network slices.

Algorithm 2 An algorithm for dynamic resource sharing

$R_{X=S_{MaxS}}$ 0 ▷ Resources in max utilized slice
$R_{X=S_{MinS}}$ 1 ▷ Resources in min utilized slice
k is a defined constant and k 1
if $R_{X=SjU} > R_{X=SjUT}$ **then**
  R_{req} k ($R_{X=SjU}$ $R_{X=SjUT}$)
  while R_{req} 0 **do**
    while S in slice **do**
      if $R_{X=S_{minS}}$ $R_{X=SjU}$ **then**
        S_{minS} S
      end if
    end while
    if ($R_{X=S_{minS}jA}$ $R_{X=S_{minS}jU}$) R_{req}) **then**
      Allocate R_{req} from S_{minS} to S_S
      R_{req} = 0
      break
    else
      Allocate $R_{X=S_{minS}jA}$ $R_{X=S_{minS}jU}$ to S_S
      R_{req} = R_{req} ($R_{A;min;S}$ $R_{MinU;S}$)
    end if
  end while
else if $R_{X=SjU} < R_{X=SjLT}$ **then**
  while S in slice **do**
    if $R_{X=S_{MaxS}} < R_{X=S}$ **then**
      S_{MaxS} S
    end if
  end while
  Allocate ($R_{X=SjLT}$ $R_{X=SjU}$) to S_{MaxS}
end if

III. Feasibility Evaluation

This section evaluates the feasibility of the proposed solution. The experiment is performed using real SFs, on top of a real NS testbed.

A. Testbed Implementation

1) System Model

An L5GO-based environment that consists of 4 network slices has been considered. We assume that 25% of total resources from each slice are allocated to the security slice.

2) Implementation setup

Here, we build a test setup to compare the performance of the proposed security framework concerning the inbuilt NS security paradigm. Figure 2 shows the implementation of our test setup. NS testbed implementation in [9] is used as the base of our implementation. Standard opensource tools have been utilized in the testbed implementation.

OpenBaton is an open-source implementation of the European Telecommunications Standards Institute's (ETSI's) NFV Management and Orchestration (MANO) framework. It is used as the NFV Orchestrator (NFVO) in our implementation. Openbaton is deployed as docker containers in an Ubuntu environment. OpenStack is an open-source cloud operating system that helps to manage resources (compute, network, and storage resources) at a data centre. It is the Virtual

Infrastructure Manager (VIM) used in the implementation. OpenStack VIM driver is used to connect OpenBaton and OpenStack installations. Here NSM is implemented as a spring boot application, and APIs can be exposed to present security requirements of the slice owners. A Suricata instance, an open-source, fast, and robust network threat detection engine is used as the SF of the evaluation. Tcpreplay, which is an open-source utility program for replaying the previously captured pcap files, is used to send data packets at different speeds to the Suricata instance. We used OpenVirtualSwitch (OVS) for diverting traffic between VNFs.

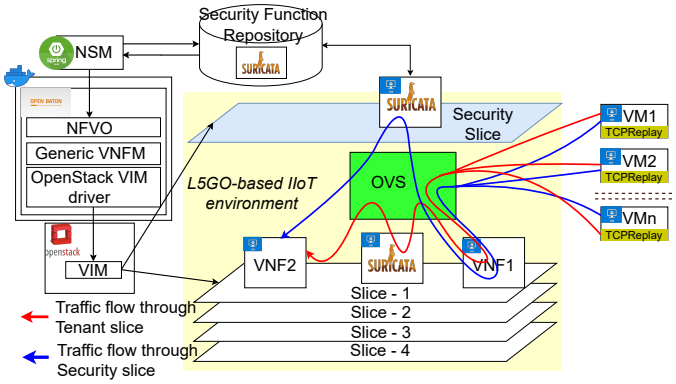

Fig. 2: Implementation setup

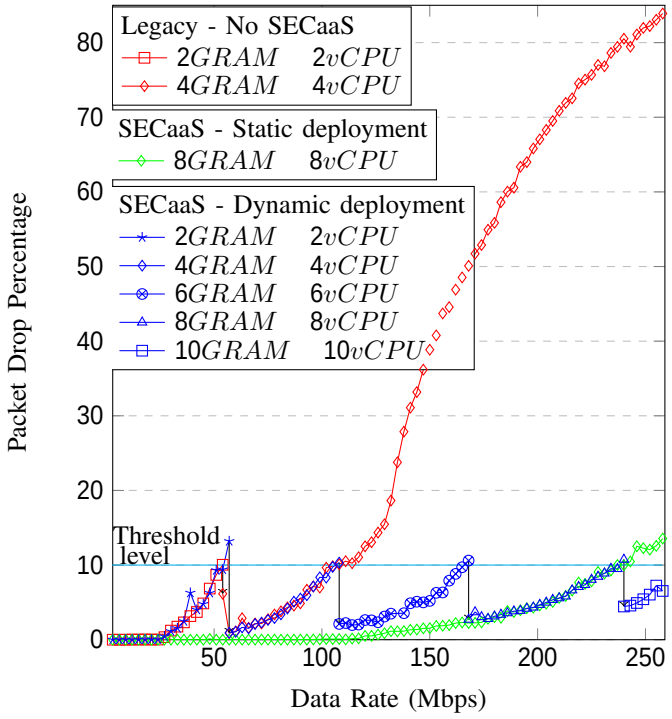

Fig. 3: Experiment Results

3) Testcase

In this paper, we consider a sample system to prove the functionality of the proposed framework. Comparison is performed for three scenarios:1) without the proposed framework, 2) Proposed framework under static deployment, 3) proposed framework under dynamic deployment.

A particular slice contains 8GB RAM and 8vCPUs. In the first scenario, 2GB RAM and two vCPUs are allocated to each VNF initially. In the second scenario, resources from all slices are allocated to the security slice statically. As there is one security VNF in the security slice, all resources (8GB RAM and 8 vCU) are allocated to that SF. Resource allocation is dynamically altered in the third scenario. The experiment is run by allocating 2GB RAM and two vCPUs for the security slice initially and dynamically changing the resources according to the given algorithm.

The traffic flows in two different paths through the system are shown in figure 2. The red path shows the legacy method (without the security framework). The blue path is for the traffic flow with the security framework. It is operated under two scenarios: static and dynamic. In all traffic flow scenarios, we use Suricata VNF as our SF. Then we gradually increase the speed of the traffic (data rate) to the slices and measure the Drop Packet Percentage (DPP) for different data rates. When DPP exceeds a threshold level (here, we selected 10% as the threshold level), the NSM alters the resource distribution in the system.

Assumption: In the 'without security framework' scenario, the maximum slicing resource that can be allocated for SFs is limited by the dependencies of the tenant slice. Since the considered usual slice has 3 VNFs, the maximum resource limit for the Suricata instance in the tenant slice is 4GB RAM and 4vCPUs. In the dynamic SECaaS scenario, 2GB RAM and 2vCPUs are allocated in each step. Furthermore, we assume that the NSM monitors the performance matrices of the VNFs (here, the DPP of Suricata instance) and allocate resources appropriately according to the monitored information.

4) Evaluation of the results

Figure 3 depicts the results of the experiment. In the first scenario, DPP increases with the increasing data rate in the Suricata instance in the tenant slice and passes the threshold level. Once the NSM increases the resource allocation of the Suricata instance, the DPP falls below the threshold level and starts again to increase with the data rate. However, the NSM can not reallocate resources to the Suricata instance due to the resource restriction in the tenant slice. Hence, DPP passes the threshold level and continuously increases.

As all the resources in the security slice are allocated to the security VNF in the second scenario, DPP can be maintained under the threshold for a wide range of data rates. However, due to the static resource allocation for the security slice, after 240 Mbps, the threshold level can not be maintained.

When we increase the data rate in the third scenario, the DPP increases as in the first scenario. As the resource allocation to the security slice is dynamic in this, we can see saw teeth in the graph. In each sawtooth, when DPP passes the threshold level, the NSM can allocate more resources to the security slice from other slices. Hence, the DPP of the Suricata instance can be controlled under a desirable threshold value.

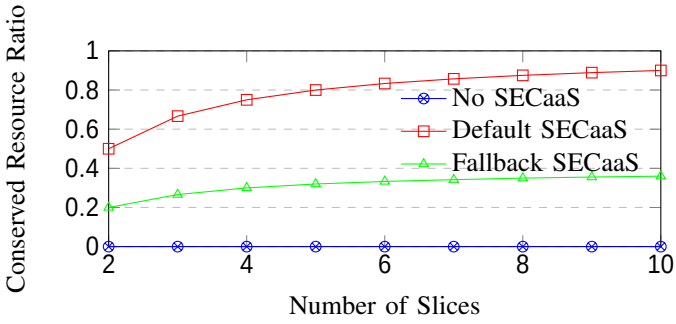

Fig. 4: Conserved resources with different modes

We can conclude that the system performs well with SECaaS than the traditional architecture. In the static SECaaS scenario, after a particular point, the system can not provide security requirements with existing resources in the system. However, in the dynamic scenario, security requirements can be provided with already existing resources in the system.

B. Simulations

We compare two modes of operation in the simulations: default mode and fallback mode, along with the traditional mode(i.e. no security framework implemented). We consider resources, cost, and latency in the simulations. Matlab is used as the simulation tool in these simulations.

1) Simulation model

We consider an L5GO environment where we can increase the number of network slices in the system. As in the experiment, we assume that 25% of the resources of a slice are allocated for security solution deployment. In default mode, 25% of resources from one particular slice is sufficient for deploying SFs in the security slice. In fallback mode, we assume that 10% of resources are required for special SFs, and the remaining 15% is sufficient for basic SF deployment. Thus, in fallback mode, 15% of resources from each slice and 10% from a particular slice are allocated to deploy SFs.

2) Resource conservation

Here, we calculate the conserved resources in different modes as follows. Modes for R_{T_P} can be default, fallback, or traditional. In traditional mode, $R_{T_T} = R_{T_P}$.

Figure 4 shows the conserved resource proportions in each mode. Conserved resources always increase when the number of slices increases in the system with the proposed framework. Since the resources need to be allocated for basic SF deployment in each slice in fallback mode, the conserved resource ratio is always a lower value than the default mode.

3) Cost reduction of the security

This simulation calculates the cost of each mode of operation. One cost unit represents 1% of allocated resources. For instance, allocating 25% of resources for security operations represents 25 cost units for security operations. The cost for each mode is calculated as follows.

$$Cost_{traditional} = N \quad cost\ for\ security\ functions\ in\ one\ slice \tag{11}$$

$$Cost_{default} = cost\ for\ security\ functions\ in\ one\ slice \tag{12}$$

$$Cost_{fallback} = N \quad cost\ for\ basic\ security\ functions + cost\ for\ specific\ security\ functions \tag{13}$$

Figure 5 shows the results received for this simulation. The cost for deploying SFs in No SECaaS mode continuously increases with the number of slices in the system. SECaaS in fallback mode always has a lesser cost value than in No SECaaS mode due to the requirement of deploying basic SFs in each slice in traditional method. Since all the SFs are deployed only in the security slice, the cost is minimum and the same for deploying security in default SECaaS mode.

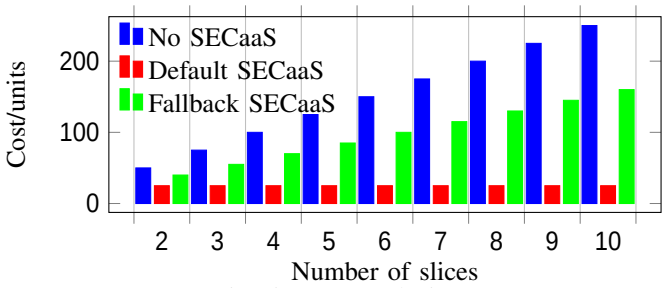

Fig. 5: Cost Analysis

4) Latency

The latency variation in the system according to the data rate is modelled here. The considered parameters are shown in figure 6 and the equations are shown below.

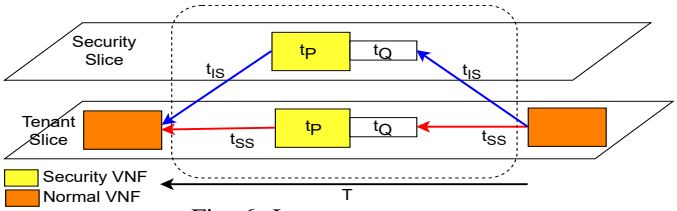

Fig. 6: Latency parameters

$$T_{No\ SECaaS} = t_{SS} + t_P + t_Q + t_{SS} \tag{14}$$

$$T_{D\ SECaaS} = t_{IS} + t_P + t_Q + t_{IS} \tag{15}$$

$$T_{F\ SECaaSs} = (t_{SS} + t_P + t_Q + t_{SS})j_{dr=o\ to\ dr=k_o} + (t_{IS} + t_P + t_Q + t_{IS})j_{dr=k_o\ to\ dr=1} \tag{16}$$

The parameters for this simulation are obtained from the above experiment. Until 20mbps, there are no packet drops in traditional mode. Therefore, packets do not need to wait in the queue for processing. Therefore, $t_Q = 0$ from 0 to 20 Mbps. After 20 Mbps, t_Q increases proportionally to the incoming data rate. We assume that $t_{SS} = t_p/10 = t_{IS}/2 = t$ in this simulation. Poisson distribution is utilized to randomize the input traffic to the system.

Figure 7 depicts the results for latency variations in different modes of operation. Initially, latency is high in default SECaaS mode. However, it can maintain the same latency for higher

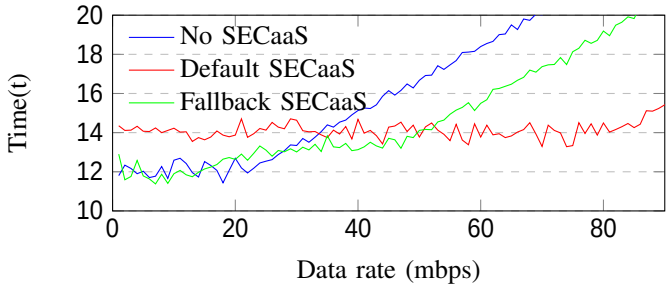

Fig. 7: Latency variation in the system

data rates. Though the traditional mode shows low latency values for low data rates, latency increases rapidly after a certain value. Fallback mode behaves in the middle of default and traditional modes in terms of latency.

IV. Discussion

Here, we compare our framework with existing solutions and present the implementation challenges of our framework.

A. Comparison with Existing work

Table II shows a comparison between our framework and existing frameworks for evaluating a set of features. 'Y' denotes the availability of the corresponding feature, and 'N' denotes the unavailability.

Feature	[5]	[6]	[7]	[8]	[3]	Our system
Security	Y	Y	Y	N	N	Y
Scalability	Y	N	Y	Y	Y	Y
Dynamicity	Y	N	Y	Y	Y	Y
Flexibility	Y	Y	N	Y	Y	Y
Network slicing	Y	Y	Y	N	Y	Y
L5GO specificity	N	N	N	Y	Y	Y

TABLE II: Comparison with existing frameworks

This comparison shows the significance of our proposed solution. Security frameworks that are L5GO specific and NS based are not proposed yet. Moreover, in terms of other advantages, our solution performs well.

B. Implementation Challenges

Implementation of the proposed framework raises several challenges. Slice isolation is a primitive requirement in the NS ecosystem to preserve the confidentiality of the traffic flow. However, traffic flow is required to steer to the security slice and back to the parent slice in the proposed security architecture. Hence, slice isolation is a challenge. In the traditional scenario, there is no specific element to manage security VNFs. The novel architecture consists of a security slice and an SFR. Hence, an additional security manager is required to manage the migration of SFs to the security slice and alter the resource allocation to the SFs in the security slice. The extra security manager will be an additional overhead to the system. In the traditional scenario, an inter-slice traffic steering mechanism is not required. Hence, the E2E latency values will be lower than the E2E latency values in the proposed system. This can be a bottleneck for delay-critical applications.

As this architecture is proposed for an L5GO-based environment, complete slice isolation is not a critical requirement. However, advanced encryption mechanisms such as homomorphic encryption [10] can be utilized to ensure secure inter-slice communication. The role of the extra security manager can be integrated into the operation of the traditional NSM to eliminate the challenge of the requirement of an additional security manager. A hybrid mechanism can be proposed for applications to mitigate the latency challenge. For delay-critical applications, they can use in-slice SFs to achieve security requirements; for other applications, the novel SECaaS architecture can be integrated.

V. Conclusion

Natural limitations in the NS-enabled private 5G network ecosystem intensify a network-level solution for addressing security vulnerabilities. In this paper, we introduced the concept of allocating a dedicated slice of the network to enable the SECaaS paradigm for an NS-enabled L5GO ecosystem. The proposed security slice performs all the related security operations in the private 5G slices. The extensive experimental and simulation results show the significance of the proposed solution. Identified implementation challenges open a set of future research directions related to the proposed framework. Finally, this paper concludes on the fitness of the concept of the security slice to address security vulnerabilities in the private 5G network domain.

References

[1] M. Matinmikko-Blue, S. Yrjölä, V. Seppänen, P. Ahokangas, H. Hämmäinen, and M. Latva-Aho, "Analysis of spectrum valuation elements for local 5g networks: Case study of 3.5-ghz band," IEEE Transactions on Cognitive Communications and Networking, vol. 5, no. 3, pp. 741–753, 2019.

[2] Y. Siriwardhana, P. Porambage, M. Ylianttila, and M. Liyanage, "Performance analysis of local 5g operator architectures for industrial internet," IEEE Internet of Things Journal, vol. 7, no. 12, pp. 11 559–11 575, 2020.

[3] I. Badmus, M. Matinmikko-Blue, and J. S. Walia, "Network slicing management technique for local 5g micro-operator deployments," in 2019 16th International Symposium on Wireless Communication Systems (ISWCS). IEEE, 2019, pp. 697–702.

[4] (2018, dec) What is security as a service? a definition of secaas, benefits, examples, and more. [Online]. Available: https://digitalguardian.com/blog/what-security-service-definition-secaas-benefits-examples-and-more

[5] S. Wijethilaka and M. Liyanage, "Security orchestration framework for federated network slicing."

[6] A. Thantharate, R. Paropkari, V. Walunj, C. Beard, and P. Kankariya, "Secure5g: a deep learning framework towards a secure network slicing in 5g and beyond," in 2020 10th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, 2020, pp. 0852–0857.

[7] Y. Khettab, M. Bagaa, D. L. C. Dutra, T. Taleb, and N. Toumi, "Virtual security as a service for 5g verticals," in 2018 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2018, pp. 1–6.

[8] A. Rostami, "Private 5g networks for vertical industries: Deployment and operation models," in 2019 IEEE 2nd 5G World Forum (5GWF). IEEE, 2019, pp. 433–439.

[9] T. Irshad et al., "Design and implementation of a testbed for network slicing," 2018.

[10] John, "Homomorphic encryption," Aug 2019, accessed on 15.12.2021. [Online]. Available: https://www.johndcook.com/blog/2019/07/04/homomorphic-encryption/