Muhammad, Yousuf; Yan, Zheng; Wang, Mingjun; Khan, F.

Trusted Network Slicing among Multiple Mobile Network Operators

Published in:
Proceedings - 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2020

DOI:
10.1109/TrustCom50675.2020.00150

Published: 01/12/2020

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Muhammad, Y., Yan, Z., Wang, M., & Khan, F. (2020). Trusted Network Slicing among Multiple Mobile Network Operators. In G. Wang, R. Ko, M. Z. A. Bhuiyan, & Y. Pan (Eds.), *Proceedings - 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2020* (pp. 1135-1140). Article 9343054 (IEEE International Conference on Trust, Security and Privacy in Computing and Communications). IEEE. https://doi.org/10.1109/TrustCom50675.2020.00150

Trusted Network Slicing among Multiple Mobile Network Operators

Muhammad Jehanzaib Yousuf
State Key Lab of ISN
Xidian University
Xi'an, P.R.C
yjehanzaib@hotmail.com

Mingjun Wang
State Key Lab of ISN
Xidian University
Xi'an, P.R.C
mjwang@xidian.edu.cn

Zheng Yan
State Key Lab of ISN
Xidian University, Xi'an, P.R.C
Department of Communications and Networking
Aalto University, Espoo, Finland
zyan@xidian.edu.cn

Fawad Khan
National University of Sciences & Technology
Islamabad, Pakistan
fawad.khan.xdu@gmail.com

***Abstract*—5G mobile networks are expected to be much bigger in size, faster in speeds and better in scalability, providing varied services to different users and businesses in contrast to previous networks. 5G will also help enabling new business models and use cases. "Network Slicing" is a driving architectural concept for multi-tenancy. Network Slicing enables Mobile Network Operators (MNOs) to deploy different services over shared physical infrastructure, increasing inter-operator resource sharing. As 5G is still in its nascent, inter operator cooperation is an area that requires immediate attention of research. Traditional inter operator trust relationship models cannot fully comprehend the needs of 5G networks. In this paper, we propose an Intel SGX based multi-MNO cooperation scheme for trusted, dynamic and efficient network slice sharing in order to support inter-operator trustworthy collaboration. Furthermore, we developed a Proof of Concept of our proposed scheme using Intel SGX, flask framework and Docker containers. The obtained results indicate the applicability of the proposed scheme with little effect on performance.**

Keywords—5G, Network Slicing, Intel SGX, Trust

I. Introduction

5G networks are touted to provide a multitude of services to highly diverse range of new businesses and verticals with conflicting requirements. Some customers or verticals would require ultra-high speed mobile broadband while some others would require ultra-low latency reliable communications [1]. The legacy monolithic 4G network architecture cannot support such heterogeneous network requirements. 3GPP has introduced some key architectural changes in 5G network architecture, making it a Service Based Architecture (SBA) [2]. The highlight of this SBA is transformation of key Network Entities (NEs) into Network Functions (NFs), which make the network highly scalable, programmable and flexible. Software Defined Network (SDN), Network Function Virtualization (NFV) and cloud computing technologies are utilized in synergy to create logically isolated networks deployed over shared physical network infrastructure. A resulting new architectural concept is"Network Slicing" and corresponding logical networks are called "Network Slices".

Network Slicing is at early stages and a hot topic in research community. It has got tremendous attention due to increasing demands of high data rates from data hungry and resource consuming applications and use cases. Afolabi et al. [1] have discussed network slicing history, key technology enablers and current state of art in detail. Operators can adopt Network Slicing-as-a-Service (NSaaS) to provide customized cellular network services as proposed in [2]. Network Slicing has existed in different forms in previous mobile network generations as network sharing concepts such as Multi-Operator Radio Access Network (MORAN), Multi-Operator Core Network (MOCN), GWCN (Gateway Core Network), Dedicated Core (DÉCOR) and enhancements for Dedicated Core Networks (eDÉCOR).

By allowing different parties to instantiate and run a softwarized architecture, 5G becomes inherently a multi-Tenant ecosystem, whereby a Tenant refers to a user or a user group with specific access rights and privileges over shared resources. Some business models and verticals, e.g., a connected car company or a social media network has a global customer base thus requires global coverage, which is beyond the capability of a single operator. Hence, to create a uniform set of services for end-customers, these businesses would need to provide the same services supported by similar slices over multiple operator networks. In such a roaming scenario between two operators, one operator can assume the role of "Renter" and the other of "Tenant". The Tenant will deploy its own deploying network function Virtual Machines (VMs) or Containers over the Renter MNO's rented infrastructure. The MNOs are honest-but-curious as well, indicating that they are honest to build network and provide services to users, but curious about the data being exchanged among users. Here arises key issue of integrity, confidentiality and trustworthiness of user data and traffic because of the untrusted host (Renter) infrastructure. Another challenge is trusted accounting of resources utilized by the Tenant and fairness of charging done by the renter. However, little existing work handles the above issues to ensure trustworthy inter-operator network slicing.

In this paper, we propose a scheme employing Intel SGX to achieve trustworthy cooperation for trusted, dynamic and efficient network slice sharing. Furthermore, our scheme ensures confidentiality of tenant user identity, enhance network configuration and policy management by Tenant over a rented network. In the end to show the applicability of our scheme with a proof of concept implementation.

The rest of the paper is structured as follows. We review related work in Section II. Section III presents the system and security models, followed by proposed scheme in Section IV.

Section V introduces proof of concept and last section is conclusion.

II. Background and Related Work

A. Background

In a resource sharing relationship, MNOs enter into a relationship of Renter (a resource provider) and Tenant (a resource user) similar to a cloud data center and Tenant. In case of MNOs, resources are shared for creating and sharing slices over different MNOs. A Renter MNO can provide slicing services to a Tenant MNO in two different slice sharing offer types according to [3].

Offer Type 1: Renter MNO instantiates a slice on behalf of the Tenant. It provides the Tenant only a monitoring interface or some limited capabilities to configure some parameters.

Offer Type 2: Renter MNO assumes the role of an infrastructure provider and provides physical computing, storage and network resources. The Tenant deploys its own VNFs and VNF managers and orchestrators, providing connectivity between VNFs through SDN.

The trust relationship in case of these two offer types are explored below:

Trust Relationship of Offer Type 1:

The Tenant has to trust the Renter MNO that it will abide by the agreed upon Service Level Agreement (SLA) and provide the settled services to the Tenant users. The Tenant has no technical means to enforce sufficient resource allocation, confidentiality and integrity of Tenant traffic data as it is only provided with a monitoring interface or maybe not.

Trust Relationship of Offer Type 2:

The Tenant has more control over its slice in Offer Type 2, since The Tenant deploys its own network functions, so user data is not visible to the Renter. Although being an Infrastructure Provider, the Renter MNO could still access the data and traffic. In this kind of scenario, a NS (network slice) should be created in such a way that a Tenant's devices need no extra authentication from the Renter. A device authenticated by the Tenant can access the Tenant slice, including some common functions shared with MNO. Tenant has better visibility of the resources assigned to it. Despite the NF deployment by the Tenant, it still has to trust the Renter MNO share some common functions like Radio Access Network (RAN) scheduler for assignment of radio resources or have to expose critical infrastructures like Authentication Server Function (AUSF) and Unified Data Management (UDM) towards the Renter. In event of cyberattack or security breach of Renter infrastructure, these network functions will be inadvertently vulnerable to the breach.

B. Related Work

A lot of work has been done on network slice orchestration and management, resource allocation and sharing. In [4], different novel provisioning models for 3rd-party services and their isolation properties were discussed but they lack monitoring of assigned resources and their usage. Ni et al. [5] proposed a protocol to build up network slicing for IoT services in 5G networks, however, this work cannot address secure and trusted slicing establishment between multiple mobile network operators. In [6], Tian et al. proposed a C-RAN Inter-Operator Cooperation Scheme to support the cooperation of multiple operators in C-RAN and allow them to share resources in a trustworthy and secure way based on Trusted Computing Platform (TCP), but this is only limited to RAN part only.

An interesting method for securing network slice is Trust Zoning. A security trust zone is a set of configurations and policies to ensure necessary security of a trust zone. Bian et al. [7] proposed a distributed authentication and authorization solution called Trust Zone, for an edge network. This solution is specifically useful for uRLLC network slices. This work is further extended by Schinianakis et al. [8]. Based upon different services provided by different network slices and security requirements, slices are classified into different security trust zones. To discourage repeated authentication processes, a similar concept is employed by authors in [9] called Radio Trust Zone (RTZ) or Authentication Slice (AS). These RTZ or AS can encompass multiple small cells or multiple network slices. This scheme works by radio fingerprinting a device and storing this fingerprint in a centralized database of trust zone or AS. Thus even if the device roams across multiple cells, it only needs to be authenticated only once in a given RTZ. Security Trust Zoning is useful strategy to enforce edge network trustworthiness but it lacks support on Trust Relationship Offer 2. Niu et al. proposed a model to effectively measure trust degrees of different network slices based upon granularity, integrity and dynamic nature [10]. However, this work cannot address secure and trusted slicing establishment between multiple mobile network operators.

Blockchain is another emerging technology for achieving trust in distributed systems, such as slicing. In [11] authors proposed a blockchain and content centric networking system to ensure trusted and secure communications in vehicular network slices in 5G. Xie et al. [12] also proposed a blockchain based security and privacy preserving solution for SDN based 5G-VANETS to support dynamic and efficient trust management. Taking inspiration from recent Mirai botnet DDoS attacks, Boussard et al. [13] exploited blockchain and SDN, proposed a trust evaluation scheme for IoT devices connected in a slice. They utilized SDN to create isolated network slices and calculated trust values of different types of devices by classifying their previous interactions stored in blockchain. Taking into account these trust values, only those devices that can meet a threshold trust value are permitted by SDN controller to join a network. Yan et al. [16] proposed an adaptive trust evaluation and management framework for virtualized networks. It uses cloud computing for secure deployment of various trustworthy security services over the virtualized networks. In [17], Xu et al. put forwarded a trust insurance mechanism "TIM", based upon Trusted Computing Platform (TCP) to provide trust authentication and trusted cooperation among VNFs. In [18], a mechanism was proposed for secure and trustworthy interactions between network applications and controller. Depending on the behavior descriptors of applications, a trust mechanism was applied to limit application privileges to run in the network.

But in all of these works, trust and privacy issues were neglected. In most of the proposed multi-tenant solutions,

schemes and architectures, a tenant has to trust a renter and share user information with the renter for authorization and authentication purposes. The tenant has to share network configurations, flow rules and policies to the renter. Also the tenant has very limited visibility of resources allocated to it. Table I compares some previously proposed works and the trust relationship they provide. It is evident from Table I that all of them provide trust relationship in Offer Type I, but fail in Offer Type 2.

TABLE I. COMPARSIONS OF EXISTING WORKS

	Trust Offer Type 1	Trust Offer Type 2
[12]	✓	×
[13]	✓	×
[14]	✓	×
[17]	✓	×
[18]	✓	×
[19]	✓	×
[20]	✓	×
[21]	✓	×
[22]	✓	×

So the problem remains, how to establish a flexible, dynamic, efficient, secure and trusted network slice sharing relationship between multiple MNOs? The required flexibility and efficiency can be achieved by technologies like NFV, SDN and cloud computing. A secure and trusted relationship could be achieved by performing critical security processes and data exchanges in a Trusted Execution Environment (TEE). A promising technology is Intel Software Guard Extensions (SGX). Intel SGX provides a general-purpose TEE which utilizes an x86-architecutre instruction set to create secure and isolated memory partitions termed as "Enclaves". Enclaves provide confidentiality and integrity of enclaved codes and data through checksums and encryption [19]. In addition to protection, the enclaves also provide local and remote attestation services. Local attestation is used by enclaves deployed over a same machine to attest each other, where as remote attestation enables enclaves to authenticate the enclaves deployed over remote machines [20]. In [21], Goltzsche et al. proposed an Intel SGX based architecture for outsourcing middle box functions through VPN to securely execute inside enclaves on client machines. Similarly, in [22] authors proposed a privacy preserving architecture to maintain privacy and integrity of code and traffic, limiting the information exposure to an untrusted third party.

III. SYSTEM MODEL AND SECURITY MODEL

A. System Model

Fig. 1 shows two mobile network operators MNO_1 (Home Network) and MNO_2 (Visited MNO) with their different network slices created over same physical infrastructure. The example three network slices showed in Fig.1 namely are: (1) enhanced mobile broadband (eMBB) slice, (2) ultra-reliable low latency communication (uRLLC) slice and (3) IoT slice. All three network slices are utilizing the same underlying physical resources yet providing services with varied quality of service parameters to different users and verticals. It also illustrates a scenario where a UE_1 (user equipment) belonging to MNO_1 (Home Network) has moved into a coverage area of MNO_2 (Visited MNO). The UE_1 was previously enjoying services provided by MNO_1 (Home Network) but after exiting the coverage area or geographical boundary of its home network, it is out of service. To deliver services to its users or customers and to keep them connected, MNO_1 agrees upon a renter-tenant agreement or SLA with another network operator MNO_2. Under this agreed upon SLA, MNO_2 acting as a renter, will provide resources to MNO_1, which is in this case a tenant, to create its slice over its physical infrastructure to provide services and access to UE_1 and other such users who would migrate into the coverage area of MNO2.

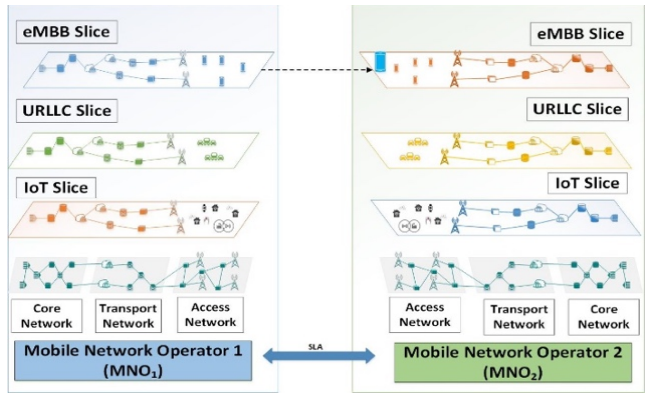


Fig. 1. System Model

B. Security Model

We assume two threat scenarios to protect against:

1. Although MNO_1 has deployed its network functions over MNO_2 physical infrastructure and MNO_2 might appear trustworthy to MNO_1 but it is not. MNO_2 is curious about the MNO_1 user information and traffic generated by it.
2. In another scenario, we assume that a powerful adversary has taken over control of the MNO_2 infrastructure and network stack including hypervisor. The MNO_1 containers are also vulnerable to this adversary. This adversary can access MNO_1 user identity and can also manipulate network configurations and misdirect traffic routes.

IV. PROPOSED SCHEME

In this section, we first overview the proposed scheme. Then, we introduce the main techniques used for deploy it. In particular, we specify UE authentication and registration protocol in inter-operator network slicing.

A. Scheme Overview

Fig. 2 shows our proposed system architecture. UE_1 belongs to MNO_1 (Home/Tenant MNO) and is accessing the network services through RAN_2 and Core network CN_2. MNO_1 deploys its necessary core network functions coupled with Intel SGX enclaves over MNO_2's (Visited/Renter MNO) rented hardware infrastructure in a dynamic way when needed. The sensitive user information e.g. user slice type or slice subscription etc., is accessed through the Intel SGX enclave created on MNO_2 rented infrastructure and afterwards it is placed in an encrypted file

system to reduce the load on enclaves and clear up enclave memory. When an MNO_1 subscriber tries to access shared network slice, its authentication and authorization process is done by MNO_1's own network functions. The NFs communicate with Home network through SGX enclave. SGX enclave provides necessary privacy protection for subscriber data and network function code. Note that SGX enclave running at MNO_2 has been previously attested and authenticated by MNO_1 as trusted and can authenticate and serve MNO_1's UE, as well as securely communicate with MNO_1 functions.

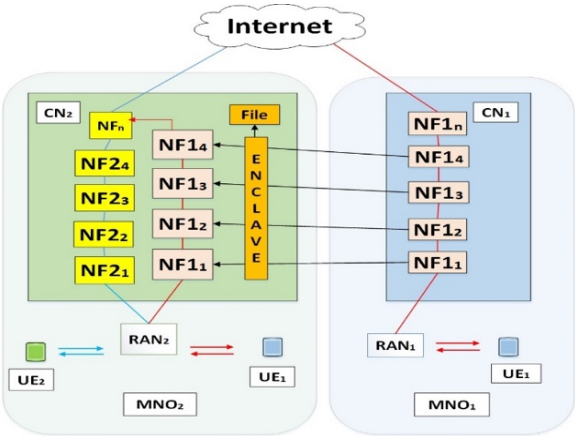


Fig. 2. Proposed Architecture

For creating and sharing a network slice, Tenant (MNO_1) requests slice sharing to Renter (MNO_2). Renter will analyze the request and its available resources. If Renter has enough resources to fulfill the requested QoS requirements from MNO_1, it will share R&B (resource and billing) details for requested slice with Tenant. If Tenant accepts the resource and billing R&B details, it will share an acknowledgement with Renter. In return, Renter will send an SLA. If Tenant accepts the SLA, it will reply with acknowledgement to Renter. Renter asks Tenant MNO_1 to share NF container images to create a slice for Tenant. After receiving NF container images, Renter deploys them over its infrastructure to create the slice, so the Tenant's users will start getting services from the created slice. During this slice sharing, both MNOs will monitor the slice KPIs (key performance indicators) and utilization of resources. Since Tenant has shared its own containers so it has the direct visibility of its slice. If any of both the parties violates the SLA, the opposite party will cancel the agreement to maintain the integrity of its network. When the time period for slice sharing is completed, the Renter will send the final billing and cost details incurred to Tenant (MNO_1). The Tenant can verify the billing by comparing with its own logs.

B. Main Techniques

For the two threat scenarios mentioned in previous section, firstly, MNO_1 needs to deploy its own network functions over MNO_2 to have effective monitoring and management. Secondly, MNO_1 needs to establish a Trusted Execution Environment (TEE) to exchange confidential UE data with its network functions in MNO_2. To create a TEE, we employ Intel SGX enclaves. The TEE of Intel SGX enclaves offers security and privacy protection for UE data through encryption and limited exposure. VMs can be a natural choice for network function deployment over remote hardware to offer secure and isolated execution due to virtualization. But at the same time, VMs can be quite beefy in size. VMs require to run complete Guest OS (Operating Systems) with pre-installed target applications, on emulated hardware. Hardware emulation is achieved by means of another software called Hypervisor or VM manager. The Hypervisor or VM manager is installed on a Host OS on underlying physical hardware. On the other hand, Linux containers are gaining traction in industry [23]–[26]. The containers neither have nor require a complete guest OS instead, they have binaries and libraries stacked into an image file called container image. This image becomes a fully functional container when it starts running. The containers are deployed in clusters. In place of Hypervisor or VM manager, a container engine for example Docker or Kubernetes is installed on Host OS, to orchestrate and manage the containers [27],[28]. The main motivations of implementing increased containers is their ease in encapsulation, deployment and isolation of applications, lightweight operations, efficient and flexible resource sharing. The size of a container is usually within megabytes (MB) while that of a VM can take several gigabytes (GB). A container usually takes less hardware resources since it does not need maintain an OS to run the same application as compared to VM. Thus, we select the container to implement our scheme for achieving better economy and efficiency.

To create a TEE, we employ Intel SGX enclaves at the Renter. Inside the NF containers, Intel SGX enclaves are established. The sensitive user data is accessed and exchanged between Tenant MNO's home NFs and remotely deployed NFs in these enclaves. The TEE of Intel SGX enclaves offers security and privacy protection for UE data through encryption and limited exposure. The NF-containers for example Docker containers can communicate with each other and are interlinked with each other over bridge networks. When Docker containers are installed, three networks namely, *bridge, none and host* are created by default [29]. Bridge network is used for communication between containers deployed at the same host. All new deployed containers over a single host, if not specified are connected to the bridge network by default. Docker does not support automatic service discovery on the default bridge network. Alternative to default bridge network to resolve IP addresses by container name, user-defined bridges are used instead. User-defined bridge networks are used to control which containers can communicate with each other, and also to enable automatic Domain Name System (DNS) resolution of container names to IP addresses.

C. UE Authentication and Registration

Fig. 3 shows the protocol of UE_1 authentication and registration at a foreign MNO_2 in inter-operator network slicing.

1. UE_1 to RAN_2: When UE_1 reaches out RAN_2/MNO_2, with Registration Request message. It will provide RAN_2 with an 5G identifier either 5G-GUTI (Globally Unique Temporary Identifier) or SUCI (Subscription Concealed Identifier) that indicates it belongs to MNO_1, while rest of Registration Request includes Registration type, last visited TAI (Tracking Area identifier), Requested NSSAI (Network Slice Selection Assistance information), PDU (Protocol Data Unit) Session status, List of PDU Sessions to Be Activated.

2. RAN_2 to AMF_1/MNO_2: Based upon the UE_1 provided 5G-GUTI or SUCI, the RAN_2 directs the Registration Request to AMF_1/MNO_2. The AMF_1/MNO_2 forwards this Registration Request data to Intel SGX Enclave. Intel SGX Enclave is created by MNO_2 over MNO_1 provided hardware resource.

3. AMF_1/MNO_2 to AMF_1/MNO_1: Since the UE_1 has provided the 5G-GUTI, Intel SGX enclave invokes the UE_1 Context Transfer service operation on the AMF_1/MNO_1 (old AMF) to request the UE_1's SUPI and UE_1 Context inside Intel SGX Enclave.

4. AMF_1/MNO_1 to AMF_1/MNO_2: The AMF_1/MNO_1 as queried in step 3, responds to the Intel SGX Enclave for the UE Context Transfer invocation by forwarding the UE_1's SUPI and UE_1 Context, SMF information, S-NSSAIs and PDU Session IDs to SGX enclave. Intel SGX Enclave saves this data into an encrypted file which can only be accessed by the enclave only.

5. AMF_1/MNO_2 to $AUSF_1/MNO_1$: After context transfer, AMF_1/MNO_2 invokes the $AUSF_1/MNO_2$ for authentication of UE_1.

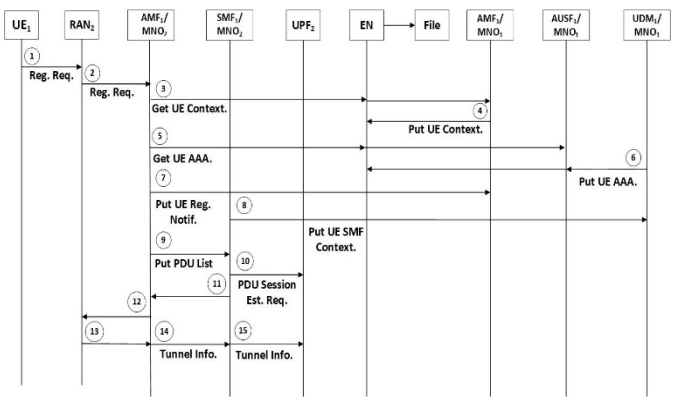


Fig. 3. UE_1 registration and PDU session establishment in MNO_1 over MNO_2 shared network slice

6. $AUSF_1/MNO_1$ to AMF_1/MNO_2: $AUSF_1/MNO_1$ requests UE_1 authentication data from UDM_1/MNO_1. After authentication, $AUSF_1$ forwards relevant security data to SGX enclave in MNO_2 and later place it in the encrypted file system. Intel SGX Enclave sends signal to AMF_1/MNO_2 that UE_1 is authenticated by MNO_1.

7. AMF_1/MNO_2 to AMF_1/MNO_1: After the AMF_1/MNO_2 has changed, the AMF_1/MNO_2 notifies the AMF_1/MNO_1 that the registration of the UE_1 in the AMF_1/MNO_2 is completed by invoking the Registration Complete Notify service operation.

8. SMF_1/MNO_2 to UDM_1: AMF_1/MNO_2 registers with the UDM_1 using UE CM-Registration service operation and subscribes to be notified when the UDM_1 deregisters this AMF_1/MNO_2. AMF_1/MNO_2 fetches the Access and Mobility Subscription data, SMF Selection Subscription data and UE context in SMF data using SDM_Get service operation from UDM inside enclave. After a successful data transfer to enclave from UDM_1, AMF_1/MNO_2 configures to be notified, when the data requested is modified through SDM_Subscribe service operation. Once again the enclave stores this SMF data in encrypted file created before for future use if required.

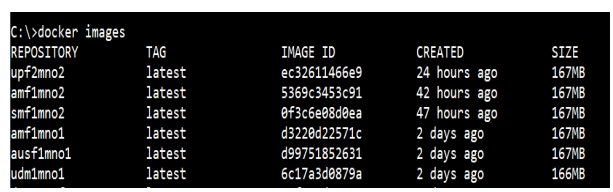
9. AMF_1/MNO_2 to SMF_1/MNO_2: Since the "List of PDU Sessions to Be Activated" is included in the Registration Request, AMF_1/MNO_2 sends PDU Session Update Session Management Context Request to enclave. The enclave notifies the SMF_1/MNO_2, associated with the PDU Sessions to activate User Plane connections for them.

10. SMF_1/MNO_2 to UPF_2: SMF_1/MNO_2 informs relative UPF_2 to establish PDU session for UE_1. SMF_1/MNO_2 selects UPF_2 and establishes session.

11. SMF_1/MNO_2 to AMF_1/MNO_2: SMF_1/MNO_2 sends PDU Session Establishment Response to AMF_1/MNO_2. The message includes the tunnel information of the UPF_2.

V. Proof of Concept Implementation

A proof of concept of the proposed scheme was developed using Python language with “Flask” micro-framework, Docker containers and Intel SGX enclaves. First, MNO_1 verifies whether the MNO_2 hardware’s Intel SGX compatibility. After verification, Intel SGX Enclave is created, which has a secure channel to communicate with MNO_1. After enclave creation, NFs are implemented as web application servers using Flask micro-framework. These web application servers are then deployed inside Docker containers. Containerization of the NFs enable their remote, dynamic and efficient on demand deployment. Fig. 4 illustrates the Docker containerized NFs deployed over shared physical infrastructure. It can be observed that Docker Containers utilize minimal memory resources that is less than 200 MB in comparison to the resource exhaustive VMs, which could consume memory resources in extent of several gigabytes (GB).



```
C:\>docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
upf2mno2            latest          ec32611466e9    24 hours ago    167MB
amf1mno2            latest          5369c3453c91    42 hours ago    167MB
smf1mno2            latest          0f3c6e08d0ea    47 hours ago    167MB
amf1mno1            latest          d3220d22571c    2 days ago      167MB
ausf1mno1           latest          d99751852631    2 days ago      167MB
udm1mno1            latest          6c17a3d0879a    2 days ago      166MB
```

Fig. 4. Docker Containers of network Functions

In order to evaluate the proposed scheme, a C++ script was written to take user data from AMF_1/MNO_1 and inject it into the enclave. If AMF_1/MNO_1 is able to map SUPI of UE_1 from 5G-GUTI provided by AMF_1/MNO_2 then the subsequent steps are performed as described in Fig. 3. After the designed steps have completed, the UE_1 is registered at MNO_2, and UE_1 data is stored in an encrypted file. If AMF_1/MNO_1 is unable to map SUPI of UE_1 from 5G-GUTI provided by AMF_1/MNO_2, it will terminate process with an error message.

Fig.5 shows the comparison between time taken by the proposed scheme with and without enclave for per UE authentication and slice selection. The process is repeated for 10 UEs. We can observe that introduction of Intel SGX enclave in authentication process induces a time cost in UE authentication and slice selection. The delay results from cryptographic function calls. For the proposed scheme, the time taken for the first UE is about 10 times higher than that without enclave based UE authentication. This relatively large difference is due to the initialization of enclave and the creation of the encrypted file.

After the first UE authentication, the time elapsed drops to almost half for any further authentications.

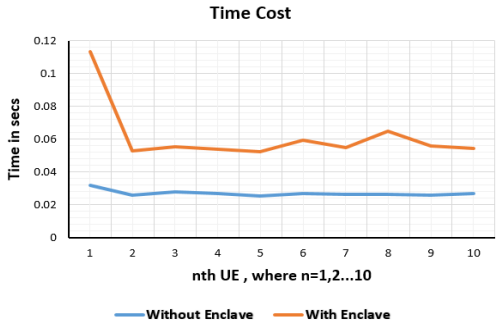


Fig. 5. Time Cost for UE Authentication Based on Enclave

VI. Conclusion

In this paper, we proposed an Intel SGX based multi-MNO cooperation scheme for trusted, dynamic and efficient network slice sharing. To showcase the applicability of our proposed scheme, we developed a Proof of Concept using Intel SGX, flask framework and Docker containers. The simulation results show that a slight effect on performance due to Intel SGX's cryptographic function calls.

Acknowledgment

The work is supported in part by the National Natural Science Foundation of China under Grants 61672410 and 61802293, the Academy of Finland under Grants 308087 and 314203, Shaanxi Innovation Team project under grant 2018TD-007, and the 111 project under grant B16037.

References

[1] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Communications Surveys and Tutorials*, vol. 20, no. 3, pp. 2429–2453, 2018.

[2] X. Zhou, R. Li, T. Chen, and H. Zhang, "Network slicing as a service: Enabling enterprises' own software-defined cellular networks," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 146–153, 2016.

[3] 5G NORMA Deliverable D3.3, "5G NORMA network architecture - Final report," 2017.

[4] P. Schneider, C. Mannweiler, and S. Kerboeuf, "Providing strong 5G mobile network slice isolation for highly sensitive third-party services," *IEEE Wireless Communications and Networking Conference, WCNC*, vol. 2018-April, pp. 1–6, 2018.

[5] J. Ni, X. Lin, and X. S. Shen, "Efficient and Secure Service-Oriented Authentication Supporting Network Slicing for 5G-Enabled IoT," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 644–657, 2018.

[6] F. Tian, Z. Yan, X. Liang, and P. Zhang, "Trusted cooperation among virtual base stations in C-RAN," *IEEE Access*, vol. 6, pp. 57787–57801, 2018.

[7] B. Han, S. Wong, C. Mannweiler, M. Dohler, and H. D. Schotten, "Security Trust Zone in 5G networks," in *2017 24th International Conference on Telecommunications (ICT)*, 2017, pp. 1–5.

[8] D. Schinianakis, R. Trapero, D. S. Michalopoulos, and B. G. Crespo, "Security Considerations in 5G Networks: A Slice-Aware Trust Zone Approach," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 2019, pp. 1–8.

[9] C. M. Moreira, G. Kaddoum, and E. Bou-Harb, "Cross-Layer Authentication Protocol Design for Ultra-Dense 5G HetNets," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–7.

[10] B. Niu, W. You, H. Tang, and X. Wang, "5G network slice security trust degree calculation model," *2017 3rd IEEE International Conference on Computer and Communications, ICCC 2017*, vol. 2018-Janua, pp. 1150–1157, 2018.

[11] V. Ortega, F. Bouchmal, and J. F. Monserrat, "Trusted 5G Vehicular Networks: Blockchains and Content-Centric Networking," *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, pp. 121–127, 2018.

[12] L. Xie, Y. Ding, H. Yang, and X. Wang, "Blockchain-Based Secure and Trustworthy Internet of Things in SDN-Enabled 5G-VANETs," *IEEE Access*, vol. 7, pp. 56656–56666, 2019.

[13] M. Boussard, S. Papillon, P. Peloso, M. Signorini, and E. Waisbard, "STewARD:SDN and blockchain-based Trust evaluation for Automated Risk management on IoT Devices," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 841–846.

[14] V. N. Sathi, M. Srinivasan, P. K. Thiruvasagam, and S. R. M. Chebiyyam, "A Novel Protocol for Securing Network Slice Component Association and Slice Isolation in 5G Networks," pp. 249–253, 2018.

[15] H. Yang, Y. Li, S. Guo, J. Ding, Y. Lee, and J. Zhang, "Distributed Blockchain-Based Trusted Control with Multi-Controller Collaboration for Software Defined Data Center Optical Networks in 5G and Beyond," in *2019 Optical Fiber Communications Conference and Exhibition (OFC)*, 2019, pp. 1–3.

[16] Z. Yan, P. Zhang, and A. V. Vasilakos, "A security and trust framework for virtualized networks and software-defined networking," *Security and Communication Networks*, vol. 9, no. 16, pp. 3059–3069, Nov. 2016.

[17] G. Xu, Y. Tang, Z. Yan, and P. Zhang, "TIM: A Trust Insurance Mechanism for Network Function Virtualization Based on Trusted Computing," 2017, pp. 139–152.

[18] A. L. Aliyu, P. Bull, and A. Abdallah, "A Trust Management Framework for Network Applications within an SDN Environment," in *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 2017, pp. 93–98.

[19] F. X. Simon P Johnson, Uday R Savagaonkar, Vincent R Scarlata and and C. V. R. McKeen, "Technique for supporting multiple secure enclaves," 2015.

[20] Anati, S. Gueron, S. P. Johnson, and V. Scarlata, "Innovative Technology for CPU Based Attestation and Sealing," 2013.

[21] D. Goltzsche *et al.*, "END BOX : Scalable Middlebox Functions Using Client-Side Trusted Execution," in *48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2018, pp. 386–397.

[22] S. Kim, J. Han, J. Ha, T. Kim, and D. Han, "SGX-Tor: A Secure and Practical Tor Anonymity Network With SGX Enclaves," *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2174–2187, 2018.

[23] "HAProxy - The Reliable, High Performance TCP/HTTP Load Balancer." [Online]. Available: http://www.haproxy.org/.

[24] "Redis." [Online]. Available: https://redis.io/.

[25] B. Fitzpatrick, "Distributed caching with memcached," *Linux Journal*, vol. Volume 2004, no. Issue 124, p. 5, 2004.

[26] W. Reese, "Nginx: The High-performance Web Server and Reverse Proxy," *Linux J.*, vol. 2008, no. 173, Sep. 2008.

[27] M. H. Syed and E. B. Fernandez, "A reference architecture for the container ecosystem," in *Proceedings of the 13th International Conference on Availability, Reliability and Security - ARES 2018*, 2018, pp. 1–6.

[28] "What are Containers and their benefits | Google Cloud." [Online]. Available: https://cloud.google.com/containers/.

[29] Docker Inc, "Docker container networking." [Online]. Available: https://docs.docker.com/v17.09/engine/userguide/networking/.