Shielding Network Function on a Multi-Operator System using SGX

FINSE May 9, 2018

Enio Marku

Joint work with Gergely Biczok

Norwegian University of Science and Technology

Outline

- Introduction
- Motivation
- Methodology
- Current work
- Future work

Introduction

- Phd at Norwegian University of Science and Technology
- Place: Trondheim
- Department: Information Security and Communication Technology (IIK)
- Supervisor: Colin Alexander Boyd
- Co-supervisor: Poul Einar Heegaard

About me

- I hold bachelor degree of Electronical Engineering from Polytechnic University of Tirana (UPT)
- I hold master degree in Telematics and Informatics from Czech Technical University in Prague (CVUT)
- My master thesis: “ Implementation of PIR protocol and deployment in Amazon Cloud”
- I worked as software developer for 2 years for New Era company located in London.
- I have started my Phd on September 2017

Methodology

- Investigation of multi operator systems in order to use it to support new 5G services
- Deep understanding of NFV, SFC and Intel SGX technologies
- Look in previous research related in outsourcing NFV to a third party provider
- Analyze previous schemes which aim to secure communication in a multi operator system

Motivation

- 5G will extend, personal communication and video services with the integration of cloud, IoT and machine to machine communication by adding new verticals

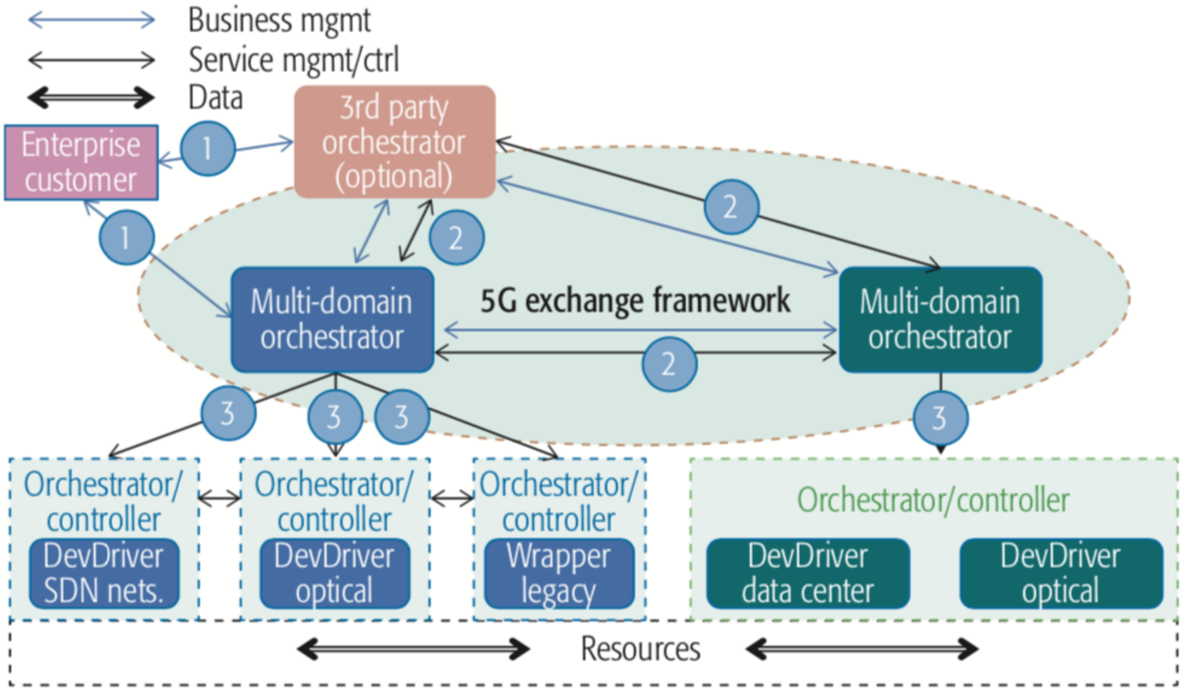
- The nature of these verticals are very demanding in economical and technical terms

- Multiple network, cloud and connectivity provider stakeholders constitute the multi-actor value chain of 5G services which inevitably require multi operator business

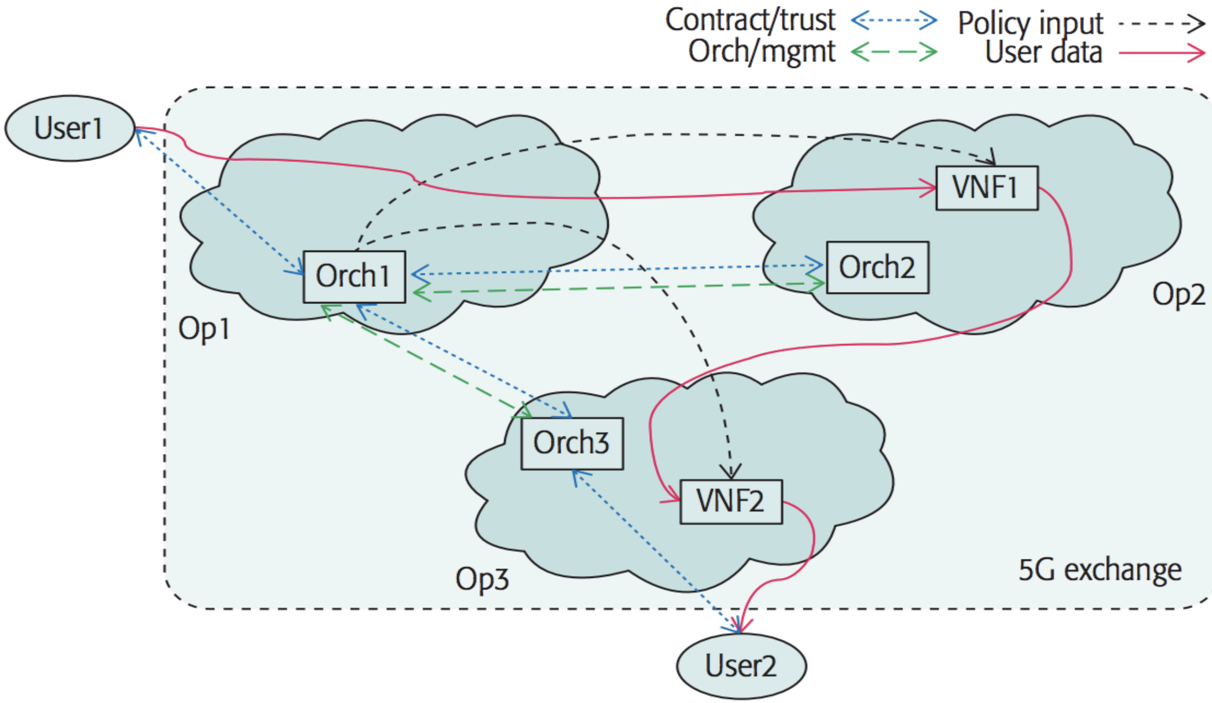
Motivation

- MOS have been implicitly supported over the internet by means of pure connectivity and interconnection agreements between operators
- Existing peering and transit interconnection agreement do not fulfill the requirements needed to support new verticals
- One general-purpose technology to use for 5G services is IPX
- NFV show promises for being a key enabler in this context
- On the other hand outsourcing NFV come with a price: Security

Simplified conceptual architecture

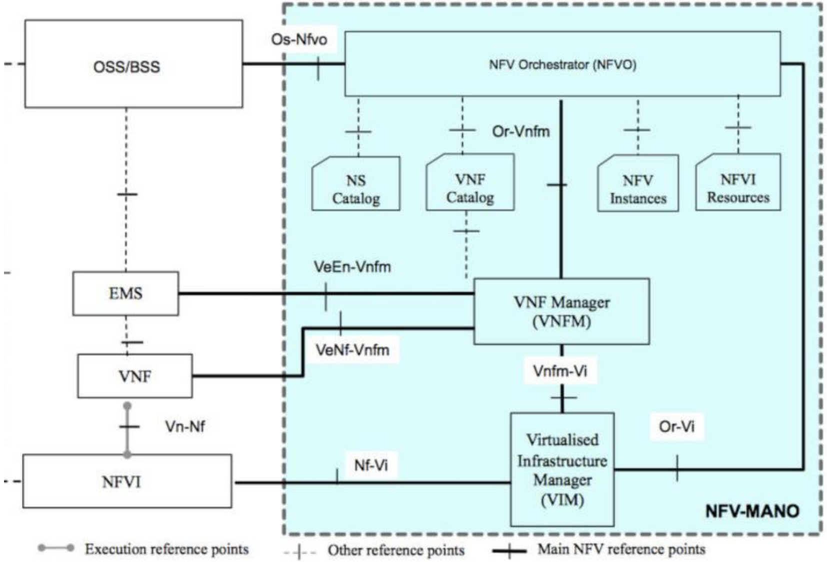


Information flows in 5GEx scenario



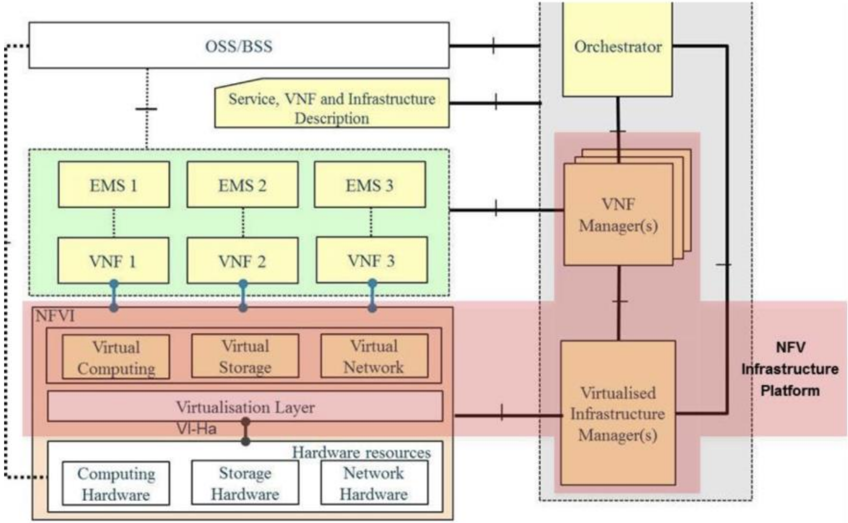
NFV Software

- It is used to deploy NF to create cloud based application with combination of virtualization software and industry standard hardware
- NFV software applies to a group of technologies being deployed by networking vendors and service providers.
- Using NFV software has major benefits such as reduced capital expenditure (capex), reducing operating expense, increase management efficiencies
- One example is NFV-MANO



NFV Orchestration

- It is used to coordinate the resources and networks needed to set up cloud based services and application
- This process use a variety of virtualization software and industry standard hardware
- Cloud service providers or global telecom operators use NFV orchestration to quickly deploy services, or VNF
- Service coordination, service monitoring, service chaining, scaling services are the requirements of Orchestration



Security concerns to deal with…

- Protect client traffic from operators
- Protect traffic from NF
- Protect NF source code
- Protect policy inputs

Homomorphic encryption

- Is a form of encryption that allows computation on ciphertext, generating an encrypted result which, when decrypted, matches the result of the operations as if they had been performed on the plaintext.
- The purpose of homomorphic encryption is to allow computation on encrypted data
- Partially homomorphic encryption

    1. Unpadded RSA
    2. ElGamal
    3. Goldwasser-Micali
    4. Damgard-Jurik

Homomorphic encryption

- Somewhat fully homomorphic encryption (FHE)

  1. Brakerski-Gentry-Vaikuntanathan
  2. Brakerski's scale-invariant cryptosytem
  3. The Gentry-Sahai-Waters cryptosystem

- Use cases

  1. Cloud computing
  2. Multi operator system
  3. Bitcoin Split-Key Vanity Mining

Possible attacks on homomorphic encryption

- FHE scheme (if they exist) seems to be highly secured, however some attacks can be performed

- In a paper with authors Yupu Hu and Fenghe Wang they managed to perform an attack on FHE scheme
    1. Construct a modified secret key
    2. Construct a modified decryption algorithm
    3. Construct a subset of the ciphertext

- In a paper with authors Zhenfei Zhang, Thomas Plantard they have performed Reaction Attack over a FHE

Is homomorphic cryptographic approach good for our scenario?

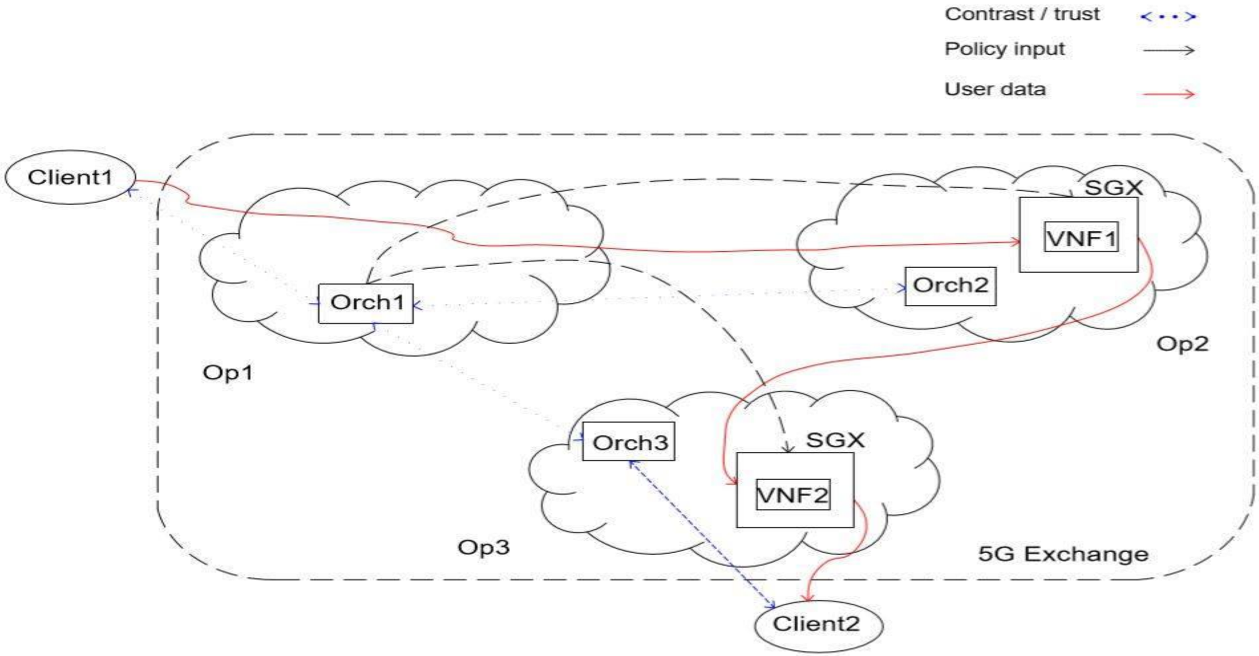
- Both, Yes and No
- Why is it a good choice

  It seems to fulfill our security requirements to a certain extent

- Why is it not a good choice

  1. Most of homomorphic protocols are not fully homomorphic
  2. Extremely slow
  3. Capacity storage increase

Overall architecture

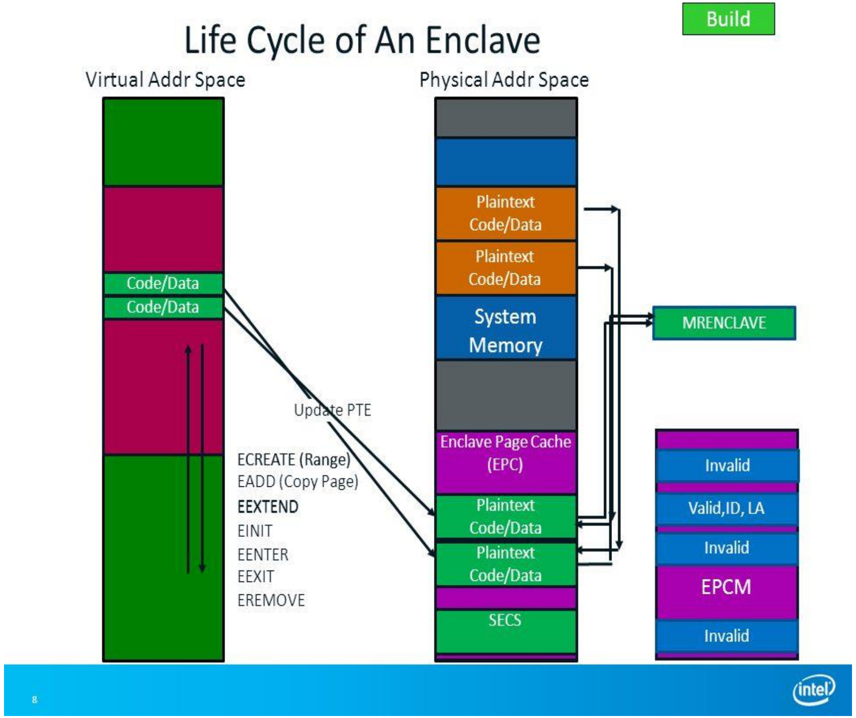


Intel-SGX

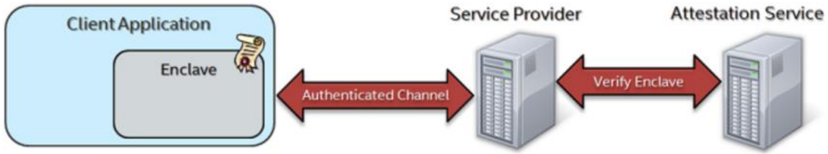
- SGX is a set of CPU instruction codes from Intel that allow user-level code to allocate private regions of memory called enclaves
- Enclaves protect from processes running at higher privilege levels
- Hardware protect Enclave area and prevent access from things that are outside of the Enclave space

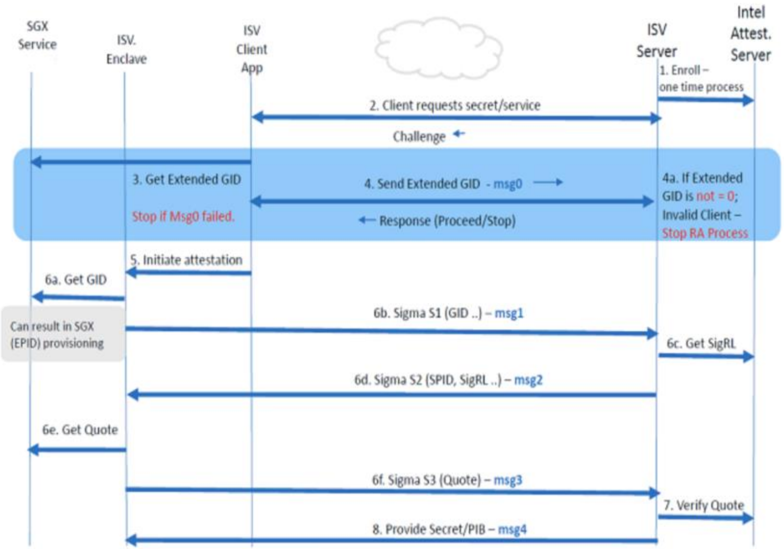
Life cycle of an Enclave

- ECREATE (create a range inside VAS which is going to be part of the Enclave)
- EADD (copy page)
- EEXTEND (cause measurement register to be updated)
- EINIT (declare that Enclave is executable)
- EENTER (Is used to enter the Enclave)
- EEXIT(Is used to exit the Enclave)
- EREMOVE (It remove all pages)



SGX Remote Attestation: How it works?




SGX Service
ISV. Enclave
ISV Client App
ISV Server
Intel Attest. Server
1. Enroll – one time process
2. Client requests secret/service
Challenge
3. Get Extended GID
Stop if Msg0 failed.
4. Send Extended GID - msg0
Response (Proceed/Stop)
4a. If Extended GID is not = 0; Invalid Client – Stop RA Process
5. Initiate attestation
6a. Get GID
6b. Sigma S1 (GID ..) – msg1
Can result in SGX (EPID) provisioning
6c. Get SigRL
6d. Sigma S2 (SPID, SigRL ..) – msg2
6e. Get Quote
6f. Sigma S3 (Quote) – msg3
7. Verify Quote
8. Provide Secret/PIB – msg4

SGX Remote Attestation: How it works?

- Allows a remote client system to cryptographically verify that specific software has been securely loaded into an enclave
- It uses CPU-based attestation
- When a client requests remote attestation, the enclave generate a report signed by the processor
- This report contains a hash measurement of the enclave
- The enclave can also bootstrap a secure channel with the client by generating a public key and returning it with the signed report

Threat model

- Abstract Enclave assumption

  1. The attacker can not observe any information about the protected code and data in the enclave
  2. Remote attestation establish a secure connection between correct parties and loads the desired code inside the enclave

- Attacker Capabilities

  1. We have considered an attacker which can compromise the software stack of the operator outside the enclave
  2. This kind of model implies, the attacker can observe communication between hardware enclaves as well as communication on the network

Threat Model

- Network function

  1. Each NF is trusted only with the permission given to it by the enterprise for specific packet fields
  2. For instance if the client give a NAT read/write permission for the IP header, the NF is trusted to not leak the header to unauthorized entities and to modify it correctly

Open issues

- How to partition the code stack of NF application
- How to avoid transitions between enclave and non-enclave code
- How to isolate NFs which are running in chain
- What framework to use for the development of arbitrary NFs

Thank you for your attention!

Questions?