Providing a Sliced, Secure, and Isolated Software Infrastructure of Virtual Functions Through Blockchain Technology

Gabriel Antonio F. Rebello, Gustavo F. Camilo, Leonardo G. C. Silva, Lucas C. B. Guimarães, Lucas Airam C. de Souza, Igor D. Alvarenga, and Otto Carlos M. B. Duarte

Universidade Federal do Rio de Janeiro - GTA/COPPE/UFRJ

***Abstract*— Network slicing, network function virtualization (NFV), and software defined network (SDN) technologies provide agile on-demand end-to-end services. The identification of a faulty virtual function becomes mandatory because services allocate resources across a distributed and trustless environment composed by multi-tenant competing service providers. In this paper, we propose and develop a blockchain-based architecture to provide auditability to orchestration operations of network slices and to provide secure VNF configuration updates while ensuring isolation and privacy between network slices. A proof of concept prototype using the Hyperledger Fabric platform was developed in which network slice runs on an isolated channel. The results show that we can secure a network slice creation, but that the consensus and the number of transaction required by the slices are a great challenge.**

I. Introduction

Mobile networks provide a connectivity model with multiple network services tailored to meet the demand of each customer segment. Software-defined networking (SDN) and network function virtualization (NFV) are the main technologies that enable virtualization to provide network programmability. Therefore, NFV and SDN technologies create an end-to-end network function chain (SFC) [1] to provide on-demand services tailored for each application. Even though the association of NFV and SDN provides the agility and low cost desired by telecommunications, it also brings new security challenges [2]. Furthermore, the impact of possible attacks increases because attacks on network function hosts may simultaneously compromise thousand of users. Therefore, it is of major importance to reduce the possible VNF attack vectors and to provide secure and reliable configuration management. Ensuring isolation between network slices is essential to avoid common attacks in shared infrastructures. In addition, tenants from each slice share the same cloud infrastructure, and chains could involve virtualized functions instantiated in competing providers domains. The multi-tenant and multi-domain environment increases the possibility of attacks inside the cloud, while hindering accountability of service providers when a fault occurs. Therefore, we must ensure that the service chain is built in a trustful manner in a trustless environment.

We argue that auditability is mandatory to identify a faulty or compromised VNF configuration, and that blockchain technology provides the required characteristics of non-repudiation and immutability of previous configuration history. We propose to use blockchain technology to register all commands that create, modify, configure, migrate, and destroy the network functions of each network slice as signed transactions. Therefore, all network malfunctions can be verified and an error could be correctly accounted to the network provider in the trustless multi-tenant competing environment.

In previous papers, we evaluate the performance of using blockchain in network function virtualization (NFV) for securing management commands, updates, and migration of virtual network functions while ensuring anonimity [3], [4].

In this paper, we focus the use of blockchain in network slicing. Network slices support requirements for delay-tolerant vehicular networks, Internet of Things, Industry 4.0 and critical services such as e-Health, smart cities, and smart grids. The extraordinarily diverse scenario requires different blockchains with specific characteristics adapted to the required service. Hence, we propose to address different slice requirements through different categories of blockchains. The blockchain data structure, the consensus protocol, and the communication protocol are tailored to each specific network slice functionality. We present a blockchain architecture for creating secure network slices tailored for each end-to-end use case. We develop and evaluate a prototype of our architecture with different types of blockchain using the open-source Hyperledger Fabric [5] platform. The prototype implements two smart contracts, Hyperledger chaincode, with specific transaction formats for protecting network slice management and VNF configuration operations. Each network slice runs in an isolated Hyperledger channel. The results show that we can secure the network slice construction, but that optimized data structures are required for scaling the number of transactions required by the slices.

II. Related Work

Several works explore the blockchain state of the art applied to communication network problems and 5G [6], [7], [8], [9], [10], [11], [12]. They focus on the use of blockchain as a replicated incremental data repository in which all past transactions are signed and recorded with asymmetric cryptographic keys. Yahiatene *et al.* and Ortega *et al.* leverage the use of blockchain as a mechanism to provide security in vehicular networks [6], [7]. They argue that blockchain can provide authenticity and trust with low latency to enable secure networks. Thuemler *et al.* and

Capossele *et al.* discuss 5G requirements for enabling e-Health based on real world experiences [8], [9]. The paper results show that blockchain provides the necessary privacy and data protection for securing medical records in a trustless environment. Rawat *et al.* propose a blockchain-based solution for virtual wireless networks that enables trust in cloud providers [10]. Rosa and Rothemberg provide guidelines to incorporate blockchain-based Distributed Applications into multiple administrative domain scenarios [11]. Boudguiga *et al.* present a solution for updating IoT devices through blockchain-stored information [12]. Our proposal provides a blockchain slice taxonomy that encompasses and generalizes all aforementioned blockchain applications as network slicing use cases.

Other works investigate the problem of security vulnerabilities in multi-tenant and multi-cloud NFV environments [13], [14]. They show that trust in cloud providers is uncertain and that compromising a single VNF at the network core endangers entire end-to-end services. Zaowat and Hasan proposed SECAP, a blockchain-based framework to securely store a provenance tree of cloud applications [15]. The framework protects the logs of application state changes. Bozic *et al.* propose an architecture for managing execution states of virtual machines using a blockchain-based system [16]. The system uses a blockchain structure to log the virtualization hypervisor instructions as transactions.

Concerning network slicing security, Bordel *et al.* propose a solution for providing intra-slice security for IoT devices and base stations in 5G systems based on pseudo-random number generators [17]. Khettab *et al.* propose to use NFV and SDN technologies to secure multi-domain network slices by dynamically instantiating security network functions such as firewalls and intrusion detection systems [18]. These works, however, do not address possible malicious behavior from network administrators.

Other works propose the use of blockchains to ensure trust in network administrators and slice brokers. Valtanen *et al.* analyze the use of blockchain in 5G network slice brokers for resource gathering, configuration, and allocation in industrial automation [19]. The paper points out the advantages to use blockchains in 5G use cases. Backman *et al.* propose to use blockchain for managing virtualized 5G network resources in a multiple-administrated scenario [20].

III. Securing Network Slicing with Blockchain

Blockchain is a replicated data structure that ensures the trust and proper functioning of a distributed system without the need for a common central authority. A representation typical of blockchain is shown in Figure 1. A value resulting from a cryptographic hash function identifies each block and each block contains both the transactions performed in a given time interval and the identifier of the previous block. In this distributed structure, each node participating in the consensus has a local copy of the blockchain that contains all transactions from the beginning. Since the current copy is the same on any network node, the non-repudiation property between members is guaranteed because all transactions are signed and each member uses its public key as identification.

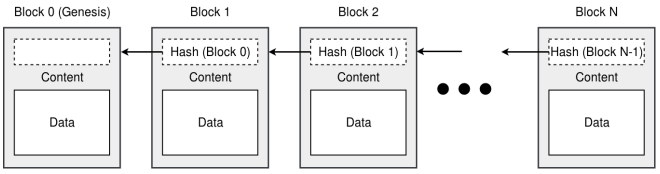

Fig. 1. Structure of a blockchain in which each block is associated with the following block and the cryptographic hash function ensures the integrity of each block.

Blockchain utilization is required in distributed environments where participants are unable to agree on a centralized authority that governs all sensitive network procedures.In virtualized data centers in which multiple cloud services are orchestrated, the presence of a malicious VNF in a service can affect the entire chain through which packets are routed without the knowledge of the cloud administrator. Furthermore, if an attacker has access to the orchestrator, the operations log can be manipulated to hide a threat. The use of blockchains, although involving a greater amount of processing as a whole, allows managing and updating VNFs in a distributed and secure way, where transactions can be checked locally by each node with the guarantee of non-repudiation and integrity.

A. Attacker Model

We consider a Dolev *et al.* attacker model, that is, the attacker is able to read, send, and discard a transaction addressed to the blockchain, or any packet of the network [21]. The attacker may either passively connect to the network and capture message exchanges or actively inject, replay, filter, and exchange information. The attacks can target tenants, VNFs, the blockchain itself, and the network.

Blockchain attacks attempt to prevent a legitimate transaction or block from being appended to the blockchain. In order for a blockchain attack to succeed, the attacker must control a significant portion of the network to affect the consensus algorithm. A fault-tolerant consensus protocol mitigates this type of attack. Attacks that require transaction corruption or tampering are impossible when every transaction includes its correspondent signed hash.

Attacks on tenants or VNFs consist on attempting to obtain configuration information or impersonate the target. Personification attacks are not possible when every transaction sent to the blockchain is signed by its issuer. Sensitive information encryption mitigates attacks that seek to obtain configuration information, where the attacker needs to obtain the private key of the targeted recipient. Our work does not address the case in which a tenant or VNF is compromised through terminal invasion or key hijacking. The proposed architecture, however, provides for the absence of any active listening service in a VNF and for the use of a terminal in read-only mode, hence mitigating attack vectors.In addition, the proposed architecture permits auditing of all past transactions at will. Therefore, if an attacker tries to modify the blockchain using stolen key pairs, the attempt will be logged. Upon discovery of an incident, the tenant can easily replace

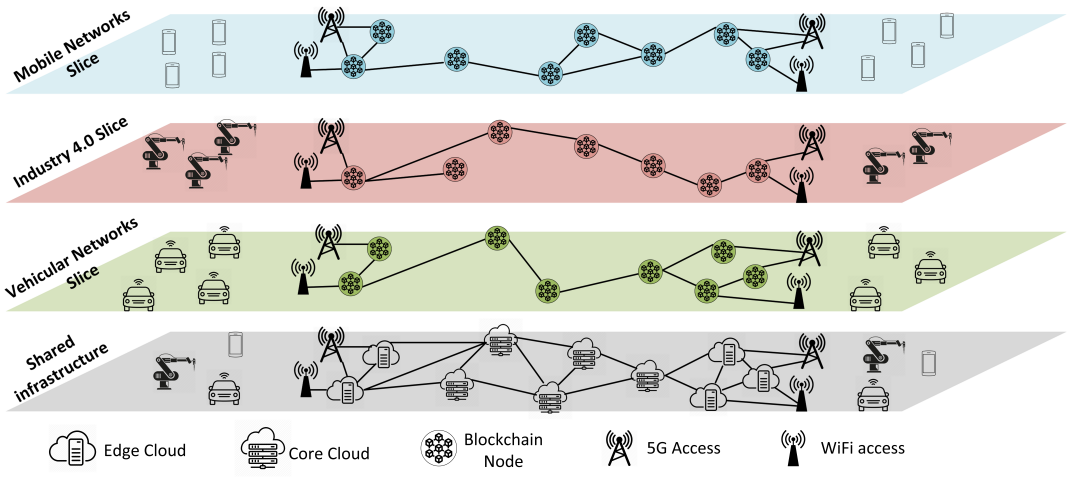

Fig. 2. Isolated network slices through blockchain in a shared physical infrastructure. Each network slice is adapted to the needs of each scenario.

the stolen key pairs, reestablishing security and preventing further damage.

Network attacks represent the attempt to isolate a single tenant, a group of tenants or a group of VNFs from the network, thus preventing the network from performing transactions or reading content from the blockchain. This attack category contemplates classic network attacks, which can be mitigated by establishing redundant paths between the distributed blockchain and VNFs or tenants. We assume that a redundant public network, e.g. the Internet, interconnects all the participants. The aforementioned assumption hardens single entity targeting if the attacker is not in its adjacent network. A complete mitigation of network attacks is outside the scope of this work. In the proposed architecture, we focus on blockchain attacks and anticipated transactions. Nevertheless, by eliminating listening services in VNFs, our architecture eradicates application-layer denial-of-service attacks, which are a common threat in shared cloud environments.

IV. The Proposed Architecture

Different use cases require specific functionalities and incur in different blockchain characteristics for each network slice. Instead of attempting to address every use case, we propose classes of blockchains that fit many use cases. Thus, a simple blockchain slice taxonomy composed of four categories of blockchain-powered slices can address a plethora of possible cases:

- Single administrator slices. This category fits use cases in which the entire network slice is administrated by a single entity. Blockchains act merely as a distributed database in which the decisions of the network are controlled by a central authority, as in private networks.
- Multi-domain crash fault tolerant slices. This category fits use cases in which individual nodes in the network could fail but the network is ultimately safe from malicious behaviour. This category provides high efficiency for multi-domain decentralized environments that ensure network security through contract-based policies. This is ideal for networks of up to tens of administrators. Examples include horizontal communication and agreement between SDN controllers in order to implement a service. This kind of slice is powered by federated/consortium blockchains using the RAFT or PAXOS protocols.
- Multi-domain byzantine fault tolerant slices. This category uses byzantine fault tolerant (BFT) consensus protocols to protect the network from malicious behavior. BFT protocols provide reasonably high efficiency for environments with up to a few hundred identified nodes. Use cases include multi-domain and multi-tenant NFV-enabled environments that implement end-to-end services. This category is powered by federated/consortium blockchains that are robust to collusion attacks.
- Fully decentralized public slices. This type of network slice provides scalability of thousands of nodes by sacrificing efficiency and throughput. Such slices rely on proof-based protocols that determine the global truth through eventual consensus. Proof-based network slices provide high scalability as they do not need to know every node in the network to obtain consensus. Hence, this type of slice is better suited for public networks with many devices, such as IoT slices.

We propose a blockchain architecture in which each category of blockchain slice addresses one or more 5G use cases, creating isolated networks with security and trust. Figure 2 depicts a scenario that uses blockchain for three network slices: a mobile network slice, an Industry 4.0 slice, and a vehicular networks slice. To ensure justice in consensus, each data center may host at most one blockchain node per blockchain slice. Blockchain nodes in a slice are invisible to anyone outside the slice. Additionally, we propose to provide auditability of slice creation and management by logging all VNF orchestration operations on a management blockchain. The management blockchain logs orchestration operations that create or modify a network slice. Every operation is signed by the client that requested the modification. The participants of the blockchain must validate each operation through consensus and provide an irrefutable signed proof that the transaction was accepted before the operations are

carried out. The signed request combined with the permanent record provided by the blockchain ensures that a malicious behaviour is traceable. Hence, the management blockchain proposal ensures provenance, accountability and traceability of faults in a multi-tenant and multi-domain NFV environment. Furthemore, we propose the use of smart contracts to provide automation and transparency in distributed trustless environments instead of trusting a particular node to receive and process transactions. The automation and transparency properties of smart contracts are ideal for creating secure end-to-end network slices that chain VNFs in multiple competing domains because they ensure that every node in the network obeys the same set of rules an that the executed code is visible to any participant node.

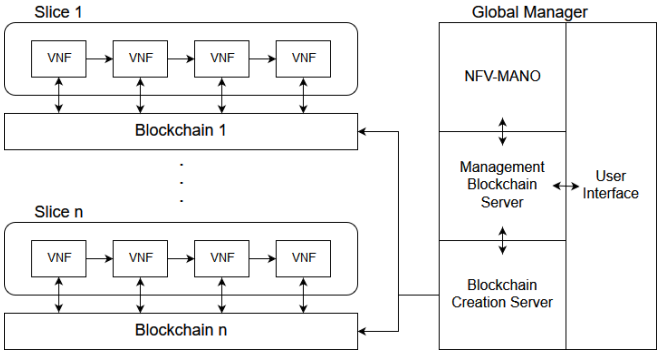

Fig. 3. The proposed blockchain-based architecture for network slicing. The user interacts with the Global Manager module in order to create secure network slices. Every VNF in a network slice is connected to a blockchain responsible for recording configuration requests and relevant information as specified by the user.

The proposed architecture, depicted in Figure 3, comprises four main components: a user interface, the NFV management and orchestration (NFV-MANO) module, a blockchain creation server module, and a management blockchain server module. The modules compose the Global Manager module, which is responsible for connecting the client to the offered services. In our architecture, the client interacts with the Global Manager module through a user interface to create/modify a secure network slice or to request slice information such as VNF and chain configuration. The client specifies the slice characteristics, such as desired VNFs and positioning restrictions, and the corresponding blockchain category. The user interface module translates the specifications to slice/blockchain creation operations and submits them as signed transactions for approval in the management blockchain. After transactions are approved, the NFV-MANO module and the blockchain creation module poll the management blockchain for pending operations. The modules then execute the operations to create new secure slices and issue signed response transactions to the management blockchain. The client can then interact with the user interface module to verify the requested secure slice was created successfully.

V. The Hyperledger Fabric-based Prototype

We develop a prototype of our proposal that uses the open-source Hyperledger Fabric platform [5]. Hyperledger Fabric is an open-source platform for implementing consortium blockchains between organizations in trustless environments. Each organization keeps a replica of the blockchain and may append blocks through consensus. The modular architecture of Hyperledger Fabric allows network administrators to design systems based on isolated subnetworks that support specific blockchains. Blockchains in Hyperledger Fabric support pluggable consensus protocols that enhance customization to fit particular use cases and trust models. Blockchain designers can configure read and write permissions to create federated and private blockchain networks. Hyperledger Fabric provides a membership identity service that manages user IDs and authenticates all participants on the network to enable permissioned networks. Nodes and channels are the most important key concepts of a Hyperledger Fabric permissioned blockchain network.

Nodes represent the entities that either participate in transaction processing or keep a copy of the ledger. Hyperledger Fabric provides three types of nodes: clients, peers, and orderers. A client represents a user that submits transactions to the peers for validation and signing, and broadcasts signed transaction proposals to the ordering service. Peers are a fundamental element of the network because they execute transaction proposals, validate transactions, and record the blockchain ledger. Peers also instantiate smart contracts and keep track of the global state, a succinct representation of the latest ledger state. Orderer nodes collectively form the ordering service, which is responsible for establishing the total order and packaging of all transactions into a block using a consensus protocol. Orderers neither participate in transaction execution nor validate transactions. The decoupling of the ordering and validation functionalities improves efficiency because it allows parallel processing of each phase. Figure 4 depicts an example of a permissioned blockchain with four organizations on Hyperledger Fabric. Each organization receives client transactions and relay them to orderers after peer validation. Each organization owns an orderer, guaranteeing justice in the consensus protocol.

Different messaging paths, called channels, isolate the blockchains. A Hyperledger Fabric channel is an isolated (private) subnetwork of communication between a subset of specific network nodes for providing privacy and confidentiality to transactions. All data transmitted in a channel, including transactions, smart contracts, membership configurations and channel information, are invisible and inaccessible to any other network members. The channel functionality fits our proposal to offer custom blockchain parameters for different required services, as it grants the network administrator the possibility to establish different block and transaction formats across the network. Hence, we can use channels to offer network slices secured by specifically-configured blockchains. Transaction formats are defined in smart contracts, called chaincode in Fabric, written in Go, Node.js or Java language.

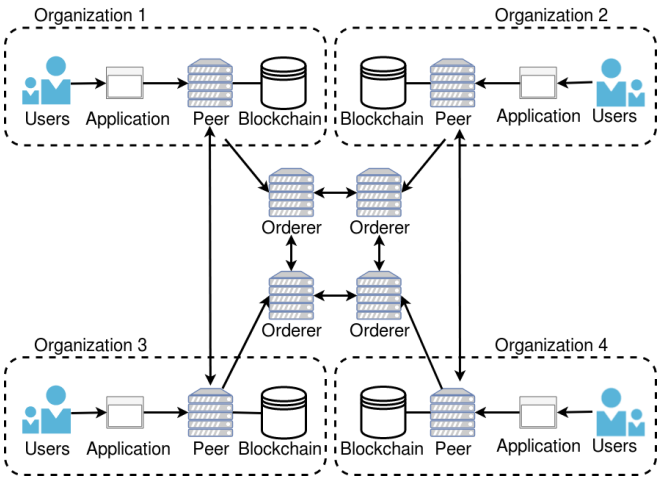

Fig. 4. A Hyperledger Fabric permissioned blockchain. Users from each organization use applications to issue transactions in the network. Peers validate transactions and relay them to orderers for global ordering and addition to a block. After a block is proposed and accepted by orderers through consensus, the peers update the blockchain and the global state.

A. Prototype Evaluation

We implement two proof of concept blockchains that represent, respectively, the management blockchain and an example of a blockchain to secure a network slice. The example blockchain secures VNF configuration update and migration on a network slice by immutably logging configurations in the blockchain. A three-organization consortium controls both blockchain implementations, by running an orderer node each. The prototype uses Kafka consensus protocol of Hyperledger Fabric to agree on transaction global order. Each of the three organizations controls one peer node that owns administration rights over the network. All three organizations receive transactions from a variable number of client nodes in the blockchain network. The slice isolation feature is provided by the Hyperledger channels through the use of TLS in every exchanged message. An Intel Core i7-7700 CPU 3.60GHz computer with 64 GB RAM creates all the nodes as Docker containers. Containers build multiple isolated user-space environments that allow the optimization of computer resources of the blockchain network.

We develop two smart contracts[1], written in Go, that run in every node of our network [3], [4]. The first smart contract, partly depicted in Listing 1, autonomously manages VNF management and orchestration through instruction and response transactions. When a client submits a slice request, the management blockchain server issues a instruction transaction with the instruction command. The contract puts the instruction transaction in a queue of pending instructions transactions. Our code notifies the NFV-MANO module, which executes pending instructions and delivers the command output to the management blockchain server. The NFV-MANO module issues a response transaction that includes a field containing the corresponding instruction transaction identifier. This provides traceability of every executed transaction in the blockchain and, hence, accountability of malicious entities.

[1]The complete code of the smart contracts is available at https://github.com/gta-ufrj/hpsr-smart-contracts

```
struct instructionTransaction
{
    command                 string
    transactionType         string
    transactionName         string
    issuer                  string
}
initialize queue
initInstruction (instruction <command,name,issuer>)
{
    if instruction is not unique or instruction is
        not well-formatted:
        return error
    putState (instruction.name, instruction)
    put (transactionID, queue)
    notify orchestrator
    return success
}
```

Listing 1. Partial pseudo-code of the smart contract for issuing instruction transactions. The command field contains the orchestration operation that the NFV-MANO must execute. We establish a queue of pending instruction transactions to be processed by the NFV orchestrator.

The second smart contract, depicted in Listing 2, sets and updates a VNF configuration. A client submits a transaction to the blockchain connected to each VNF in a network slice. The transaction contains a descriptive text for the associated configuration in the description field, as well as the configuration data in the configuration field.

```
struct configurationTransaction
{
    configurationIdentifier string
    versionIdentifier       string
    description             string
    configuration           string
    transactionType         string
    transactionName         string
    issuer                  string
}
initConfiguration (configuration <description,
    configuration,name,issuer>){
    if configuration is not unique or configuration
        is not well-formed:
        return error
    putState (configuration.name, configuration)
    return success
}
```

Listing 2. Partial pseudo-code for issuing configuration transactions. The configurationIdentifier field contains a unique identifier for a configuration.

Our prototype uses Hyperledger Fabric certificate authorities (CA) to generate and manage digital certificates of every node in the blockchain network. Digital certificates ensure auditability and that only certified and authorized nodes participate in the blockchain network.

Our prototype deploys each node of the Hyperledger Fabric network as a container in a single computer and sends transactions simultaneously. We evaluate the transaction throughput by sending an increasing number of transactions and measuring the elapsed time that the clients need to propose all transactions for the blockchain network. Figure 5 presents the results of the client transaction rate evaluation. The transaction throughput in our network slice reaches a peak value of 71.31 transactions per second on the client

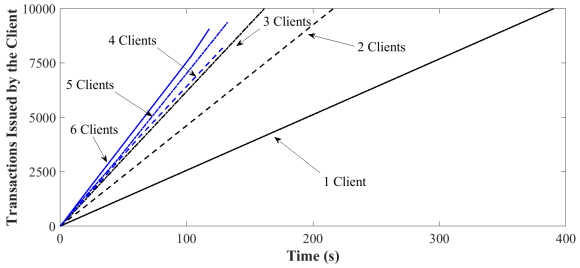

Fig. 5. Total elapsed time to process transactions in the blockchain as the number of issued transactions and the number of clients increase. The results show that the throughput increases with the number of clients issuing transactions.

side. In our experiment, we use one core for each client and, therefore, the parallelization increases the throughput.

We configure the block creation settings according to previous benchmarks performed in Hyperledger Fabric [22], [23]. We set the preferred size of the new block in bytes without the header to 99 MB, the maximum number of transactions in a block to 10, and the timeout to initialize a consensus round to 2 seconds. If any of the conditions are met, the orderer node begins a new consensus round and sends a new block proposal. After achieving consensus, the peer nodes append the new block to their blockchain copy.

VI. Conclusion

The network slicing technology provides customized end-to-end services by chaining virtual network functions between competing cloud infrastructures in a multi-tenant and multi-domain distributed trustless environment. The resulting high programmability provided by the network softwarization exposes all traffic to an increased number of threats. Therefore, it is mandatory to precisely define and locate failures and misuse to identify malicious agents that can simultaneously compromise the good behavior and quality of service of thousands of users.

We propose a blockchain-based architecture to secure the customized network slices. The plethora of different characteristics required by each slice imposes the use of various customized blockchains with different number of participants, data structures, type of transactions, transaction throughput, type of consensus, type of networks, etc.

We implement a proof of concept prototype using two blockchains that guarantee a secure slice creation and a secure network function virtualization update and migration. We use the hyperledger fabric platform that facilitates the creation of multiple blockchain in different channels. For future works, we will use optimized data structures to improve the transaction rate and use different consensus protocol.

VII. Acknowledgment

This work was financed by CAPES, CNPq, FAPERJ, and FAPESP (2015/24514-9, 2015/24485-9 and 2014/50937-1).

References

[1] J. Halpern and C. Pignataro, "Service Function Chaining (SFC) architecture," RFC7665, 2017, http://www.rfc-editor.org/rfc/rfc7665.txt. Accessed Mar. 14, 2019.

[2] A. M. Medhat, T. Taleb, A. Elmangoush, G. A. Carella, S. Covaci, and T. Magedanz, "Service function chaining in next generation networks: State of the art and research challenges," *IEEE Comm. Mag.*, vol. 55, no. 2, pp. 216–223, Feb. 2017.

[3] I. D. Alvarenga, G. A. F. Rebello, and O. C. M. B. Duarte, "Securing management, configuration, and migration of virtual network functions using blockchain," in *IEEE/IFIP NOMS*, 2018.

[4] G. A. F. Rebello, I. D. Alvarenga, I. J. Sanz, and O. C. M. B. Duarte, "BSec-NFVO: A blockchain-based security for network function virtualization orchestration," in *IEEE International Conference on Communications (ICC)*, 2019, to be published.

[5] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*. ACM, 2018, p. 30.

[6] Y. Yahiatene and A. Rachedi, "Towards a blockchain and software-defined vehicular networks approaches to secure vehicular social network," in *IEEE Conference on Standards for Communications and Networking (CSCN)*, 2018, pp. 1–7.

[7] 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.

[8] C. Thuemmler, C. Rolffs, A. Bollmann, G. Hindricks, and W. Buchanan, "Requirements for 5G based telemetric cardiac monitoring," in *14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2018, pp. 1–4.

[9] A. Capossele, A. Gaglione, M. Nati, M. Conti, R. Lazzeretti, and P. Missier, "Leveraging blockchain to enable smart-health applications," in *IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI)*, 2018, pp. 1–6.

[10] D. B. Rawat and A. Alshaikhi, "Leveraging distributed blockchain-based scheme for wireless network virtualization with security and QoS constraints," in *International Conference on Computing, Networking and Communications (ICNC)*, 2018, pp. 332–336.

[11] R. Rosa and C. E. Rothenberg, "Blockchain-based decentralized applications for multiple administrative domain networking," *IEEE Communications Standards Magazine*, vol. 2, no. 3, pp. 29–37, 2018.

[12] A. Boudguiga, N. Bouzerna, L. Granboulan, A. Olivereau, F. Quesnel, A. Roger, and R. Sirdey, "Towards better availability and accountability for IoT updates by means of a blockchain," in *IEEE EuroS&PW*, 2017, pp. 50–58.

[13] M. Pattaranantakul, R. He, Q. Song, Z. Zhang, and A. Meddahi, "NFV security survey: From use case driven threat analysis to state-of-the-art countermeasures," *IEEE Communications Surveys & Tutorials*, 2018.

[14] N. Paladi, A. Michalas, and D. Hai-Van, "Towards secure cloud orchestration for multi-cloud deployments," in *EuroSys-CrossCloud*, 2018.

[15] S. Zawoad and R. Hasan, "SECAP: Towards securing application provenance in the cloud," in *2016 IEEE 9th International Conference on Cloud Computing*, Jun. 2016, pp. 900–903.

[16] N. Bozic, G. Pujolle, and S. Secci, "Securing virtual machine orchestration with blockchains," in *CSNet'17*, 2017.

[17] B. Bordel, A. B. Orúe, R. Alcarria, and D. Sánchez-De-Rivera, "An intra-slice security solution for emerging 5G networks based on pseudo-random number generators," *IEEE Access*, vol. 6, pp. 16 149–16 164, 2018.

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

[19] K. Valtanen, J. Backman, and S. Yrjölä, "Creating value through blockchain powered resource configurations: Analysis of 5G network slice brokering case," in *IEEE WCNCW'18*, Apr. 2018, pp. 185–190.

[20] J. Backman, S. Yrjölä, K. Valtanen, and O. Mämmelä, "Blockchain network slice broker in 5G: Slice leasing in factory of the future use case," in *Internet of Things Business Models, Users, and Networks*, Nov. 2017, pp. 1–8.

[21] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[22] P. Thakkar, S. Nathan, and B. Viswanathan, "Performance benchmarking and optimizing hyperledger fabric blockchain platform," in *IEEE MASCOTS*, 2018, pp. 264–276.

[23] C. Gorenflo, S. Lee, L. Golab, and S. Keshav, "Fastfabric: Scaling hyperledger fabric to 20,000 transactions per second," Jan. 2019, https://arxiv.org/pdf/1901.00910.pdf.