B-VNF: Blockchain-enhanced Architecture for VNF Orchestration in MEC-5G Networks

Raaj Anand Mishra*, Anshuman Kalla†, Kaustubh Shukla‡, Avishek Nag§, Madhusanka Liyanage¶
*†‡School of Computing and Information Technology, Manipal University Jaipur, India
§School of Electrical and Electronic Engineering, University College Dublin, Ireland
¶School of Computer Science, University College Dublin, Ireland
¶Centre for Wireless Communications, University of Oulu, Finland
*raaj.169103057@muj.manipal.edu, †anshuman.kalla@jaipur.manipal.edu, ‡kaustubh.179302075muj.manipal.edu,
§avishek.nag@ucd.ie, ¶madhusanka@ucd.ie, ¶madhusanka.liyanage@oulu.fi

***Abstract*—The roll-out of 5G technology will nurture the realization of broadband, ultra-reliable, and zero latency services. Network Function Virtualization (NFV) and Multi-Access Edge Computing (MEC) are among the key enablers for 5G. The synergy between NFV and MEC allows migration of Virtual Network Functions (VNF) from cloud to the edge of the network thereby adding agility to the softwarized 5G networks. The overall orchestration of VNF includes, but is not limited to, processing VNF requests, selecting appropriate VNF, migrating VNF from cloud to MEC, instantiating migrated VNF at MEC, settling payment according to a VNF's usage, maintaining VNF's reputation, etc. The orchestration is not foolproof and raises doubts about its trustworthiness. To address all the existing issues in a unified manner, we leverage Blockchain technology as yet another enabling technology for MEC-enabled 5G. Thus, we propose a Blockchain-enhanced architecture for secure VNF orchestration such that issues like authenticity, integrity, confidentiality, reputation, payment transfer, and many more are resolved. To furnish a Proof-of-Concept (PoC), we develop a prototypical DApp (Decentralized Application) using Ethereum Blockchain and Suricata as an exemplar VNF. Further, we discuss the strong resiliency of the proposed architecture against numerous well-known attacks.**

***Index Terms*—5G, Blockchain, Smart Contract, NFV, MEC, VNF**

I. Introduction

5G promises extremely low latency, ultra-high speed, high reliability, and ubiquitous connectivity. There are many underlying technologies that are identified as key enablers for 5G, of which the two are Network Function Virtualization (NFV) and Multi-Access Edge Computing (MEC). NFV solves the problem of rigid and standalone (often proprietary) hardware by substituting them with virtualized and flexible software [1]. Whereas, MEC offers computational and storage facilities at the edge of the network [2].

To harness the full potential of MEC, the network needs to migrate VNFs as and when required from the cloud servers to the edge of the network. This migration and overall orchestration of VNFs are pregnable to numerous security issues, some of those are targeted in this paper (as shown in Figure 1) and are discussed below.

1) VNF Request: MEC node when sends a VNF request, the migration of VNF from cloud to the MEC happens. This process may encounter various issues such as (i) alteration of the VNF request by an intruder with malevolent intention, (ii) impersonation of MEC node to get malicious access to VNF [3], and (iii) sending fake VNF requests with intentions like getting configuration information or making DoS attack. Thus, it is important to authenticate a MEC node, verify integrity of a VNF request, and ensure confidentiality of VNF's configuration-related information.

2) Feasibility of VNF Instantiation at MEC: At times, the resources required to instantiate the solicited VNF might not be available at the requesting MEC node. The reason behind this could be a legitimate resource crunch at MEC or could be suspicious. Thus upon receiving a valid VNF request, the system must have a secure mechanism to (i) retrieve the status of currently unoccupied resources at the originating MEC node, and (ii) then compare it with the configuration requirements of the requested VNF. Absence of such a mechanism may lead to failure in VNF migration.

3) VNF Migration: During the migration of VNF, a pernicious intermediary can vandalize or tamper it. Thus, a MEC node must be provided with a secure mechanism to validate the VNF it receives before instantiating that VNF.

4) Reputation of VNF: Next, we consider third-party VNF provider [4] as an entity that develops and offers VNFs to the network operators on a payment basis. When a VNF request arrives at cloud, numerous VNFs may be available as offered by these third-party VNF providers. Possibly, these VNFs may not perform the same way as manifested at the time of advertisement. Thus the network operator requires a trusted and dynamic reputation system, based on which it can select and migrate, on the fly, the most appropriate VNF. Building such a reputation system calls for decentralized, transparent, and secure logging of performance by network elements.

5) Payment Settlement: To make payments to the third-party VNF providers: (i) usage details of migrated VNFs need to be recorded and (ii) based on these details, payment is to be made. Thus it needs decentralized and transparent tracking of the VNF's usage as well as secure and automated payment mechanism for a dispute-free business.

Our motivation is to deal with these issues in a unified

978-1-7281-7299-6/20/$31.00 © 2020 IEEE

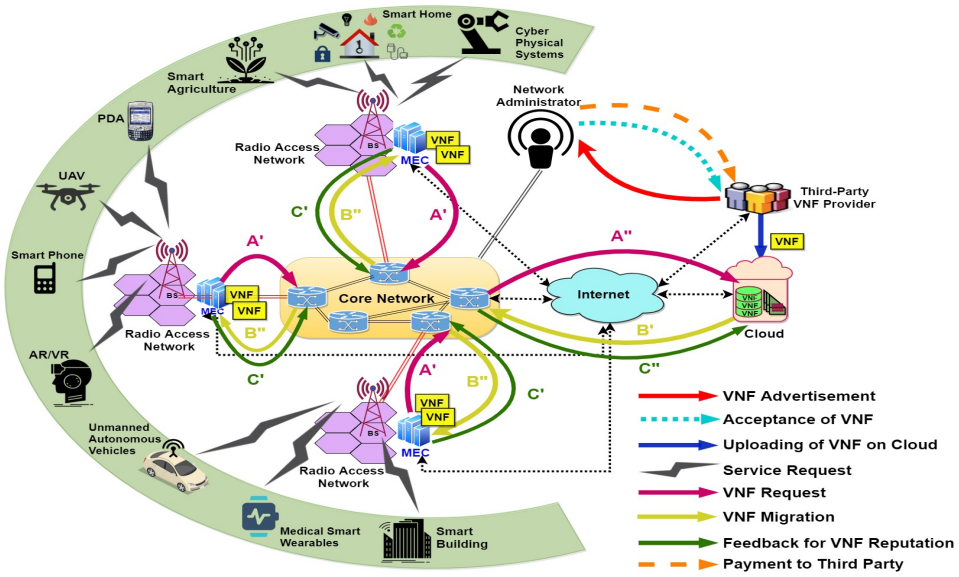

Fig. 1. Issues related to VNF migration in MEC-5G Network

manner by leveraging blockchain technology. Thus, we put forward blockchain-enhanced architecture for secure VNF orchestration in MEC-5G networks that include third-party VNF providers. To do so, we envision the blockchain network as an overlay P2P network that utilizes the underlying physical infrastructure of MEC-5G. Further, all sorts of operations (advertisements of a VNFs by a third-party provider, acceptance of a VNF by the network operator, VNF request, VNF migration, VNF performance monitoring, capturing the VNF usage details, payment to the third-party VNF provider, etc.) are considered as blockchain transactions. As a PoC, we develop a DApp that makes use of Ethereum and InterPlanetary File System (IPFS). The *demonstration and the pseudo-codes of the prototypical DApp* are available on the project's website[1].

The organization of the paper is as follows. Section II surveys existing works. Section III presents the proposed architecture. Section IV discusses the implementation. Section V discusses the results. Section VI concludes the paper.

II. Related Work

Till date, significant work has been done to overcome the impediments e.g., changes in Quality of Service (QoS) like delay, jitter, and packet loss, faced during seamless VNF migration and deployment [5]–[7]. These references mostly look for optimising migration decisions in terms of the physical network's resources, and how the migration process can be dynamically and non-disruptively addressed.

Some references like [8]–[10] establishes the importance of securing the migration of VNFs. The authors in [8] mention about different constraints in a practical network that might affect VNF migration, security being one of them. Reference [9] talks about the need for a centralized security orchestrator on top of the ETSI-NFV reference architecture. In [10], authors propose the use of OpenStack to encrypt traffic between VNFs and the delay performance for the encrypted traffic is evaluated through VPN tunnels.

Though the above references mention about security in VNF migration, none of them explicitly talk about a Blockchain-based implementation. Authors in [11] details several security threats in an NFV infrastructure and the best practices to mitigate them. While [11] gives some hints about the applicability of blockchain to solve the security issues in VNF migration scenario, references [3], [4], [12]–[15] presents different use cases of the blockchain technology for secure and trusted VNF service migration and 5G slice provisioning.

For example, [3] develops a Blockchain-based authentication framework for VNF migration but it does not consider a MEC scenario and also does not conceptualise the reputation system or the third-party payment authentication system. In [12], the creation of secure but independent slices for 5G use cases is proposed. Each slice is being secured by a different Blockchain created through the hyperledger fabric. This work also, though very much relevant to our proposition, does not mention about the authentication of VNF procurement from a third-party provider and a reputation-based VNF pricing which we are proposing. The main contribution of [13] is proposing blockchain and transaction models that provide traceability in a multi-tenant and multi-domain NFV environment. Again this work does not mention about a reputation system and a third-party based VNF-procurement model.

The authors in [4] proposes a VNF monetization framework authenticated by Blockchain. However, the business model they focus on is different. They model the problem from the end-users' perspective who would be paying an infrastructure provider (InP) to host end users' VNFs. A reverse-auction-based and Blockchain-facilitated model is proposed to ensure a fair and auditable competition between the InPs. The business model that we propose is

[1]Project's Website: https://sites.google.com/view/b-vnf-project/home

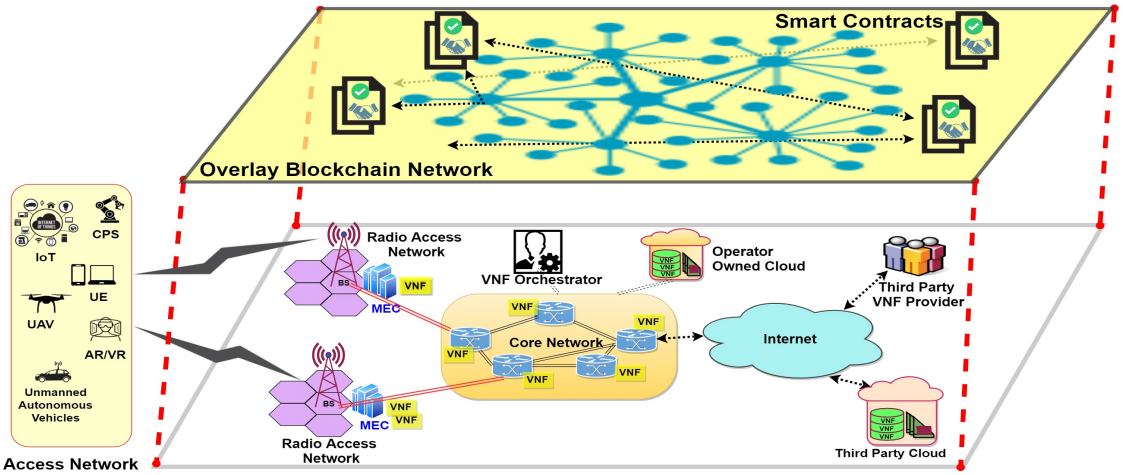

Fig. 2. Blockchain-based architecture for secure VNF orchestration in MEC-5G Networks

between the network operators/InPs and the VNF providers.

All these works from [3]- [15] are similar to different degrees to our work presented in this paper. However, the main differences in our contribution to those appearing in the above-cited references are that: (1) our approach envisages the Blockchain network as a logical overlay over the physical infrastructure where different components of the Blockchain are distributed and hosted by the servers of the physical network and any consensus-related message passing would take place using the links of the physical network; and (2) we try to integrate all aspects of the VNF migration right from the third-party provider to the network operator's cloud to the MEC nodes close to the users and build an end-to-end trust model over the entire VNF migration value chain facilitated by Blockchain.

This paper presents PoC implementation which may solve many of the challenging problems as summarised in [16]. Table I summarizes the novelty of our work compared to some of the related works. The table uses three indicators; "✓", "×", and "-" to indicate availability, absence, and not sufficient information, respectively.

TABLE I
COMPARISON OF OUR WORK WITH THE EXISTING PERTINENT WORKS

Characteristic	**Ref. [3]**	**Ref. [12]**	**Ref. [13]**	**Ref. [14]**	**Ref. [4]**	**Ref. [15]**	**Our Proposal**
Includes Third-Party VNF Provider	×	×	×	–	✓	–	✓
Reputation-System-Based VNF Pricing	×	×	×	×	–	–	✓
Authentication	✓	✓	✓	–	–	–	✓
Integrity	✓	✓	✓	–	–	–	✓
Confidentiality	✓	✓	×	–	–	✓	✓
Access Control	✓	✓	✓	–	×	–	✓

III. Proposed Architecture with Reputation System

The proposed architecture comprises of six entities as shown in Figure 2. They are (1) User Equipment (UE) in an access network, (2) MEC nodes, (3) Core network, (4) VNF orchestrator, (5) Third-party VNF provider and (6) Cloud servers (operator owned cloud and third-party cloud). As discussed in section I, the proposed architecture visualizes blockchain as an overlay P2P network. The VNF orchestrator by the virtue of the blockchain, securely oversees the entire process which involves (i) viewing the VNF advertisement sent by a third-party VNF provider and deciding to accept it or not, (ii) validating a VNF request sent by MEC node, (iii) selecting most suitable VNF from the available set of VNFs (based on the reputation) and migrating a VNF from the cloud to the MEC node that requested it, (iv) building the performance-based reputation system and (v) secure payment to third-party VNF provider. The location from where the VNF is migrated to a MEC node depends on whether the network owns the VNF or borrows it on a pay-as-you-go basis. In the former case, the VNF needs to be migrated from the operator owned cloud, however, for the latter case, it is migrated from the third-party cloud. VNFs available at the third-party-cloud are the ones that are offered by third-party VNF providers.

UE initiates a service request which arrives at the nearest MEC node. To meet the requested service, MEC node identifies the required VNF. If that VNF is already running at this MEC node then the service requested is immediately fulfilled. Else, the MEC node sends the VNF request to the VNF orchestrator. The VNF orchestrator validates the request and looks for the requested VNF. First, it looks in the operator owned cloud, if not found, then it searches the third-party cloud. It may happen that VNF orchestrator detects the presence of more than one VNFs for the requested VNF. Under this situation, the most suitable VNF is selected using the reputation system.

To ensure confidential migration of VNF (such that only the intended MEC nodes get access) the architecture utilizes the concept of the session key. A unique session key is generated which is encrypted with the public keys of all the authentic entities in the system. These encrypted session

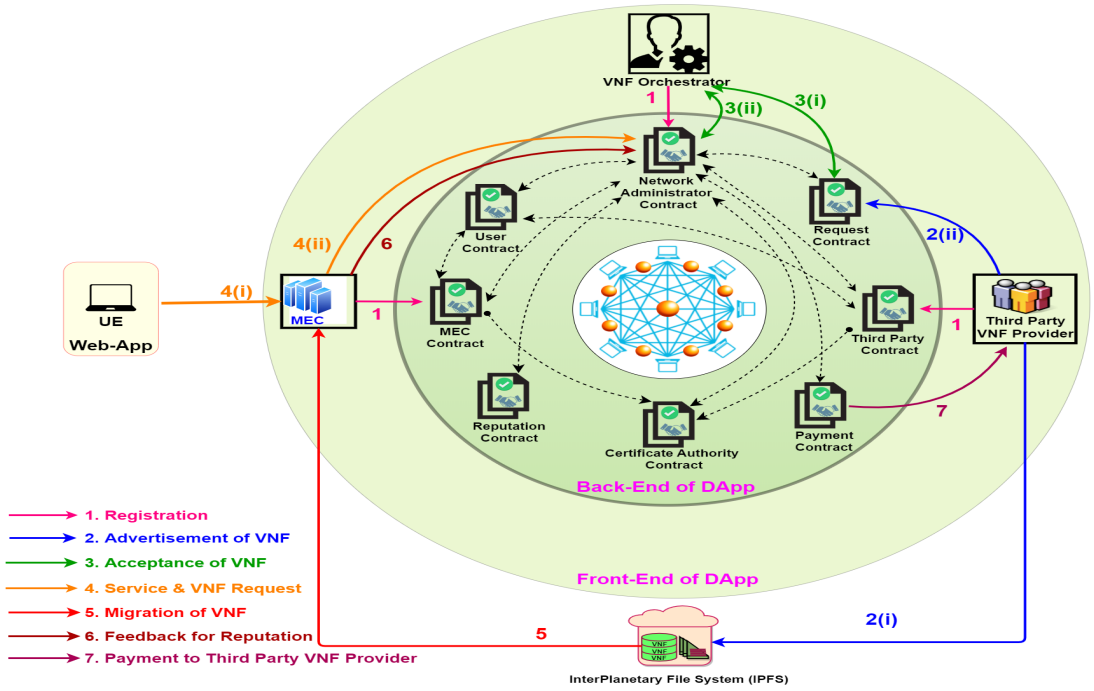

Fig. 3. Implementation of secure VNF orchestration and reputation system

keys are then uploaded on the blockchain. Thus, every entity can retrieve the respective encrypted session key from the blockchain and decrypt it using its private keys.

To build the reputation system, the MEC nodes send performance feedbacks like the status of the installation, utilization of resources, malfunctioning, etc. All of these get recorded on the blockchain. As a part of our proposal, we put forward a mechanism that works on the recorded performance and builds a reputation for every VNF. Let V be the set of VNFs, M be the set of MEC nodes, and E be the type of errors that can occur for a particular VNF at a particular MEC node. Let us define $R_v^i(t_i)$ to be the instantaneous reputation of a VNF v at time instant t_i. Let $X_{v,e}^m$ denote the number of times error $e \in E$ occurs in MEC server $m \in M$ for a particular VNF $v \in V$. The instantaneous VNF reputation is related to the number of times the VNF fails because of some error in the following way:

$$\frac{1}{R_v^i(t_i)} = \sum_{m=1}^{M} \sum_{e=1}^{E} w_e X_{v,e}^m \tag{1}$$

where w_e is the weight associated with an error type. Next, $R_v(t)$ represents the overall reputation of the v^{th} VNF, calculated at time t after every time interval T. This can then be expressed as a moving average as follows:

$$R_v(t) = \alpha \times R_v^i(t) + (1-\alpha) \times R_v(t-T) \tag{2}$$

The $0 < \alpha < 1$ represents smoothing factor. The price of a VNF C_v can be expressed as a function of its reputation as:

$$C_v = R_v(t) \times B_v \times t \tag{3}$$

where B_v is the base price advertised in the smart contract and t is the time the VNF has been used by the MEC server. Therefore, one can have a reputation based pricing for the VNFs and the price can be dynamically updated in the smart contract related to the payment. The VNFs can also be ranked based on their reputation and the historical rise and fall of their reputation based on their performance can all be recorded in the blockchain. An example plot on

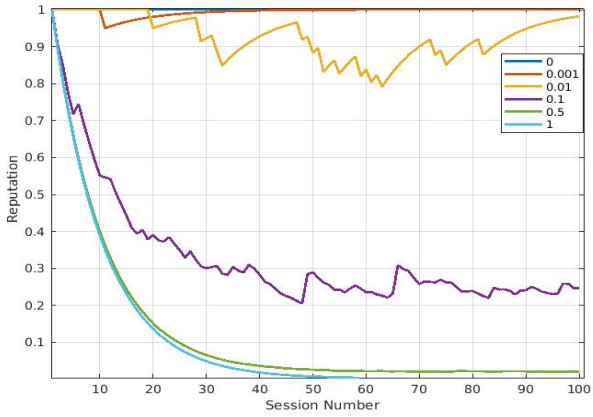

Fig. 4. VNF reputation variation over session numbers

how the instantaneous reputation of a VNF observed over the number of sessions the VNF has been utilised by a MEC server is shown in Fig. 4. It shows that the number of errors per unit time over a particular session (i.e., the values in the legend of the plot in Fig. 4) impacts the overall reputation of the VNF significantly.

IV. Implementation

The building blocks of our implementation are: (i) Ethereum platform which acts an overlay blockchain P2P

network, (ii) InterPlanetary File System (IPFS) which acts as the cloud servers, (iii) a Web application that runs over a PC and acts as a User Equipment (UE), (iv) Decentralized Application (DApp) the front-end of which runs on a PC and the back-end runs on Ethereum, and (v) Suricata is used as exemplar VNF. The developed DApp allows three different types of users; MEC node, VNF orchestrator, and third-party VNF Provider. Figure 3 depicts the logic of our implementation. Eight different small-sized smart contracts are developed in order to have better code reusability, reduced cost, and quick testing and validation .

Various software tools and libraries used for the prototypical implementation are: (i) MetaMask[2] , (ii) Web3.js[3] , (iii) aes256[4] , (iv) eciesjs[5] , and (v) ipfs-http-client[6]. Next, we discuss various steps involved in the implementation.

1) Registration: It is the bootstrap step that grants unique ID to every user. The user can be one of the three types (MEC node, VNF orchestrator, and third-party VNF provider).We have used the IDs provided by MetaMask as the unique ID. Each type of entity has a directly associated smart contract; *MEC_contract* for MEC, *Network_Administrator_contract* for VNF orchestrator and *Thrid_Party_contract* for the third-party VNF provider. For registration purpose, the directly associated smart contract in-turn calls an instance of the *User_Contract*. Moreover, a pair of Public and Private Keys are generated for each entity at the time of registration; the private key is stored in the user's device whereas the public key is stored on the blockchain using the *Certificate_Authority_contract*.

2) Advertisement of VNF: When the third-party VNF provider has a new VNF, it uploads the new VNF on to the IPFS server and in-return it gets a unique hash from IPFS. Next, it advertises the new VNF to the VNF orchestrator using the DApp which invokes the *Request_contract*. The details which are provided at the time of advertisement are a description of the VNF's functionality, system requirement to instantiate the VNF, and the pricing.

3) Acceptance of VNF: VNF orchestrator views the pending VNF advertisement by calling the *Request_contract* and decide to accept it or not. To accept, it approves the new VNF by calling the *Network_Administrator_contract* which in-turn calls an instance of *Thrid_Party_contract* to get the details like IPFS hash and system requirements (to run the VNF). It is worth to note that this IPFS hash of VNF is encrypted using a session key.

4) Service Request and VNF Request: The service request is initiated by a User Equipment (UE) (mimicked by the web-application), whereas, the VNF request is in-turn initiated by MEC node (mimicked by the DApp). Note, the VNF request is placed only if the VNF required to support the service is not already available at the MEC node. In our implementation, an automated process in DApp handles the service request sent by the web-application (i.e. UE). When the DApp receives a service request, it invokes the *MEC_contract* to find if that VNF is already running. Else, it sends the VNF request by calling the *Network_Administrator_contract*. We use Suricata as an exemplar VNF.

5) Migration of VNF: Once, the VNF request reaches the *Network_Administrator_contract*, it interacts with the *MEC_contract* to fetch the status of the currently available resources at the requesting MEC node. If the resources available satisfy the system requirements to instantiate the requested VNF, then the *Network_Administrator_contract* sends VNF's IPFS hash, encrypted with the session key, to the MEC node. To decrypt the IPFS hash, the MEC node needs the session key. So, it downloads the encrypted session key from IPFS and decrypts it using its private key to gets the original session key. Using this session key, the MEC node decrypts the encrypted IPFS hash of the VNF and fetches the VNF from IPFS. This completes the migration of a VNF from the cloud to the MEC node.

6) Feedback for Reputation: To build the reputation system, the MEC node sends two feedbacks for each VNF it uses. The first feedback is sent at the time of the successful instantiation of the migrated VNF. The second feedback is sent when the MEC node uninstalls a VNF when it is not required anymore. To send any one of the feedbacks, the MEC node calls the *Network_Administrator_contract* which uses the instance of the *Reputation_contract* and updates the reputation on the blockchain.

7) Payment to Third-Party VNF Provider: The VNF orchestrator makes the payment based on the usage, starting from the time the VNF was migrated to the time it got uninstalled. In fact, the usage details are bundled together with the two feedbacks which are sent by *MEC_contract* to *Network_Administrator_contract*. Next, the *Payment_contract* gets updated by *Network_Administrator_contract*. Thus, the VNF orchestrator views the pending payments using *Payment_contract*. Finally, the VNF orchestrator pays to the third-party VNF provider using *Payment_contract*.

TABLE II
COST OF VARIOUS SMART CONTRACTS

Smart Contracts	Transaction Cost (Gas)	Execution Cost (Gas)
User_contract	759268	535568
Network_Administrator_contract	2452110	1818026
MEC_contract	1295089	936101
Third_Party_contract	1167335	838683
Request_contract	969177	688721
Reputation_contract	607243	417455
Payment_contract	1034569	737773
Certificate_Authority_contract	618037	424461

V. RESULTS AND DISCUSSION

A. Cost Computation for Various Smart Contracts

To validate the correctness and evaluate the costs involved, various tests have been carried out on the Ethereum

[2]https://metamask.io/
[3]https://web3js.readthedocs.io/en/1.0/
[4]https://www.npmjs.com/package/aes256
[5]https://www.npmjs.com/package/eciesjs
[6]https://www.npmjs.com/package/ipfs-http-client

TABLE III
POSSIBLE ATTACKS AND COUNTERMEASURES

Attack	Description	Countermeasure by BVNF Implementation
Man-in-the-middle attack during the migration process	In such an attack, an intruder may eavesdrop, relays, or hampers the integrity (by malicious alteration) of the ongoing communication between two parties.	The prevention of man-in-the-middle attackis ensured by using smart contracts and by defining the eligible callers who can call a specific transaction/ method (for example request for migration can only be called by a MEC node). This is done by checking the digital signature of a transaction.
Broken Authentication and Session Management	Due to flaws in authentication/ session management, sensitive data and keys might be captured by an attacker.	All the users/ entities are connected to the blockchain through a verified ID (i.e., - Ethereum address). The transactions are digitally signed by the logged-in user using the verified ID, hence session can only be compromised if the user willingly gives away his account's keys.
Sensitive Data Exposure	Insecure way of handling and storing sensitive data leads to various vulnerabilities.	All the data (i.e transactions) are recorded on the blockchain as well as the VNF's IPFS hash. This maintains integrity. Further, access to the data is supervised using smart contracts, hence providing confidentiality and authenticity.
Broken Access Control	Improper implementation of access control permits an attacker to misuse the functions which may compromise the system.	Access control is maintained by the smart contracts by clearly defining the access rules for different functionalities available in the contracts.

Test Network - Rinkeby. Two types of costs that are encountered for deploying any smart contract on Ethereum are transaction cost and execution cost. The transaction cost is the gas consumed when a smart contract is sent for validation along with necessary data. Whereas the execution cost is the gas consumed for executing a smart contract and it depends on the number of variables used, the number of operations performed, and the number of function calls made. Remix[7] is used to calculate the values of both the costs. Table II exhibits the costs for various smart contracts.

B. Various Security Attacks and Countermeasures

Table III summarises some of the critical attacks, their brief description, and how BVNF implementation can countermeasure these attacks. In general, most of the attacks are being prevented by thorough validation (by the blockchain), by using public-private-key-based encryption for confidentiality and by properly structured smart contracts which are well tested on Remix[7] IDE against vulnerabilities.

VI. CONCLUSION

In this paper, we tried to address the hurdles and difficulties in the VNF migration process in a 5G architecture using the blockchain technology. A proof-of-concept prototype is proposed and tested which resolves the issues related to valid third-Party payments, MEC device and VNF authenticity, and accuracy of interpretation of whether MEC can host a definite VNF or not. We also developed a reputation system for the selection of the most appropriate VNF. A DApp is developed using Ethereum and smart contracts. The paper also provides computed cost for the deployment of this architecture alongside a mathematical explanation of the VNF reputation system. Future work will focus on finding out a scalable and resource-optimised deployment benchmark of this concept through extensive theoretical modeling, simulations, and experiments.

ACKNOWLEDGEMENT

This work is supported by Academy of Finland in 6Genesis Flagship (grant no. 318927) and European Union in RESPONSE 5G (Grant No: 789658) project.

[7]https://remix.ethereum.org/

REFERENCES

[1] "Network Functions Virtualisation (NFV)," Accessed: 20.07.2020, uRL: https://www.etsi.org/technologies/nfv.

[2] "Multi-access Edge Computing (MEC)," Accessed: 20.07.2020, uRL: https://www.etsi.org/technologies/multi-access-edge-computing.

[3] I. D. Alvarenga, G. A. F. Rebello, and O. C. M. B. Duarte, "Securing Configuration Management and Migration of Virtual Network Functions using Blockchain," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, April 2018, pp. 1–9.

[4] M. F. Franco, E. J. Scheid, L. Z. Granville, and B. Stiller, "BRAIN: Blockchain-Based Reverse Auction for Infrastructure Supply in Virtual Network Functions-as-a -Service," in *2019 IFIP Networking Conference (IFIP Networking)*, May 2019, pp. 1–9.

[5] D. Cho, J. Taheri, A. Y. Zomaya, and P. Bouvry, "Real-Time Virtual Network Function (VNF) Migration toward Low Network Latency in Cloud Environments," in *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*. IEEE, 2017, pp. 798–801.

[6] L. Nobach, I. Rimac, V. Hilt, and D. Hausheer, "SliM: Enabling Efficient, Seamless NFV State Migration," in *2016 IEEE 24th International Conference on Network Protocols (ICNP)*. IEEE, 2016, pp. 1–2.

[7] R. V. Rosa, C. E. Rothenberg, and R. Szabo, "VBaaS: VNF Benchmark-as-a-Service," in *2015 Fourth European Workshop on Software Defined Networks*, Sep. 2015, pp. 79–84.

[8] H. Ibn-Khedher, E. Abd-Elrahman, H. Afifi, and J. Forestier†, "Network Issues in Virtual Machine Migration," uRL: https://arxiv.org/pdf/1508.02679.pdf.

[9] B. Jaeger, "Security Orchestrator: Introducing a Security Orchestrator in the Context of the ETSI NFV Reference Architecture," in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1, Aug 2015, pp. 1255–1260.

[10] S. Lal, A. Kalliola, I. Oliver, K. Ahola, and T. Taleb, "Securing VNF communication in NFVI," in *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*, Sep. 2017, pp. 187–192.

[11] S. Lal, T. Taleb, and A. Dutta, "NFV: Security Threats and Best Practices," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 211–217, Aug 2017.

[12] G. A. F. Rebello, G. F. Camilo, L. G. C. Silva, L. C. B. Guimarães, L. A. C. de Souza, I. D. Alvarenga, and O. C. M. B. Duarte, "Providing a Sliced, Secure, and Isolated Software Infrastructure of Virtual Functions Through Blockchain Technology," in *2019 IEEE 20th International Conference on High Performance Switching and Routing (HPSR)*, May 2019, pp. 1–6.

[13] 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 *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6.

[14] H. Zhu, C. Huang, and J. Zhou, "EdgeChain: Blockchain-Based Multi-vendor Mobile Edge Application Placement," *CoRR*, vol. abs/1801.04035, 2018.

[15] B. Nour, A. Ksentini, N. Herbaut, P. A. Frangoudis, and H. Moungla, "A Blockchain-Based Network Slice Broker for 5G Services," *IEEE Networking Letters*, vol. 1, no. 3, pp. 99–102, Sep. 2019.

[16] A. Nag, A. Kalla, and M. Liyanage, "Blockchain-over-Optical Networks: A Trusted Virtual Network Function (VNF) Management Proposition for 5G Optical Networks," 2019.