



Transport and Telecommunication, 2024, volume 25, no. 3, 289-299
Transport and Telecommunication Institute, Lauvas 2, Riga, LV-1019, Latvia
DOI 10.2478/ttj-2024-0021

BLOCKCHAIN-BASED SECURITY FRAMEWORK FOR VNF PACKAGE PROTECTION IN 5G NETWORK SLICING SERVICES

Meriem Boukessessa¹, Abdelkader Ghazli², Adda Ali-Pacha³

^{1,3} *Laboratory of Coding and Security of Information, Faculty of Electrical Engineering, Department of Electronics, University of Sciences and Technology of Oran Mohamed Boudiaf USTO-MB
B.P. 1505, El Mnaouar – Bir el Djir, Oran, Algeria
{meriem.boukessessa, adda.alipacha} @univ-usto.dz*

² *University of Bechar – Tahri Mohamed
Rue de l'indépendance, 08000, Bechar, Algeria
ghazli.abdelkader@univ-bechar.dz*

The demand for scalable, open and granular networks has enabled mobile networks such as 5G to adopt new concepts including NFV (Network Function Virtualization). NFV separates the dependency of network functions from the hardware component, allowing them to be deployed flexibly and dynamically across network slices. In this way, operators can deliver personalized services and optimize the use of network resources, contributing to greater operational efficiency and an enhanced end-user experience. However, there are several issues to be addressed, such as the security of VNFs and the way in which service provider customers instantiate NSSIs (Network Slice Subnet Instance). In this article, we present a new approach based on blockchain 2.0, which guarantees the immutability of VNFs and templates, as well as highly secure instantiation with access management and support for replay attacks.

Keywords: NFV, Smart Contract; VNF Packages, 5G, Blockchain, Security

1. Introduction

Over the last two decades, mobile technologies have made spectacular progress, marked by constant advancements in processing power and throughput, transforming the way we communicate and access information in the most efficient way. The recent evolution of mobile technology has been profoundly marked by the implementation of 5G, which brings significant improvements in terms of throughput, latency and the ability to support more users simultaneously, thereby opening the door to new digital trends such as Augmented Reality, Autonomous Vehicles and many other innovations. 5G networks are designed to provide a flexible, scalable, agile and programmable network platform on which different services with varying requirements can be deployed and managed. These services are mainly classified into three categories, known as eMBB (Enhanced Mobile Broadband) for services focused on improving bandwidth capacity, URLLC (Ultra-Reliable Low Latency Communications) for time-critical services requiring very low latency and high reliability, and mMTC (Massive Machine Type Communications) for services aimed at supporting a large number of simultaneously connected devices (Patel *et al.*, 2023). To meet these new expectations, innovative concepts have been integrated into the 5G ecosystem considered to be key enablers, such as Network Function Virtualisation (NFV), Software-Defined Networking (SDN) and Mobile-edge Computing (MEC), transforming the way networks are deployed by operators, by using a flexible, distributed, service-based architecture that allows network functions to be delivered as separate, interoperable software components rather than on dedicated physical servers, thus replacing the traditional architecture of previous generations, which were often based on monolithic architectures. The adoption of this approach also aims to reduce the capital investment costs (CAPEX) associated with the network infrastructure, by virtualizing network functions, operators can optimize the use of hardware resources, eliminating the need to invest in dedicated equipment for each specific function, NFV aims to improve operational expenditure (OPEX) by introducing more efficient practices in the management of networks over time, through automation and orchestration, for efficient roll-out and scaling to meet demand. A Virtualized Network Function (VNF) refers to a network node that has been instantiated as a software process, such as the implementation of 5G core components, where traditionally these would be deployed

as physical devices, A VNF package encapsulates all the components information and description for the deployment of a VNF, including the software, configuration files, dependencies and any other artifacts required for the VNF to function within a network environment. As with any new technology, the benefits may also present potential security challenges if proper security measures are not considered, the VNF descriptors should have measures to guarantee the integrity of deployments to prevent any tempering or alteration of configurations, since NFV implies providers of virtualized network functions and the customers who deploy them, a secure ecosystem should be created between stakeholders, that protects against data breaches and unauthorized access (Attaoui *et al.*, 2023).

Blockchain stands as an innovative technology providing a decentralized and secure method for storing information. By incorporating smart contracts, blockchain becomes a resilient decision-support tool for managing access and authorizations or a mechanism ensuring the integrity of exchanged data. This versatility has allowed blockchain to find applications in various sectors, including the 5G security domain.

Many authors have focused on the 5G authentication process improvement using blockchain, by taking advantage of its decentralisation to overcome the risk of single point of failure. A Blockchain-based authentication solution for 5G networks framework was proposed to address the need for fast and frequent authentication. It offers advantages over existing 4G protocols by minimizing authentication processes, enhancing security, and utilizing Blockchain's features like anonymity and immutability. This solution improves authentication effectiveness, especially compared to 4G protocols, which may not handle the dynamic access requirements of 5G networks effectively (Bala *et al.*, 2022).

An efficient authentication approach was proposed for 5G networks using blockchain technology to reduce authentication overheads and keep the cost low and practical. The proposed approach is based on the 5G EAP-AKA' protocol and employs blockchain at the level of cell sites instead of access points. Authors also presented a security analysis using the CAP theorem to explain how blockchain-based authentication can help defend against DDoS attacks (Liu *et al.*, 2021).

A new framework named RUSH (Robust and Universal Seamless Handover authentication protocol for 5G HetNets) leverages the trapdoor collision property of a chameleon hash function and the global availability and tamper-resistance of a blockchain, and achieves mutual authentication, key agreement, anonymity, traceability, robustness, perfect forward secrecy, master key forward secrecy, known randomness security, key escrow freeness, and universality. Authors provided a formal security proof based on the BAN-logic and formal verification based on the AVISPA tool to demonstrate the security of RUSH against various malicious attacks. The paper also presents comprehensive performance comparisons that shows RUSH efficiency in both computation and communication (Zhang *et al.*, 2021).

(Alemany *et al.*, 2020) introduced a pioneering approach to deploying End-to-End Network Slices in a multi-domain network using an Ethereum-based Blockchain. By leveraging Blockchain technology, the authors aim to foster collaboration and efficient deployment of Network Slices across domains. The non-hierarchical architecture proposed in the paper revolutionizes the management of Network Slicing requests, offering several advantages over traditional hierarchical models.

(Mishra *et al.*, 2020) have developed and tested a Proof of Concept (PoC) called "BVNF" focusing on the security of VNF orchestration and their migration to the MEC. The system manages the announcement of a network virtualisation function as well as the validation of a VNF request by a MEC based on the latter's reputation. It then selects an appropriate VNF, migrates it from the cloud to the MEC, and performs a reputation evaluation of each MEC. The blockchain has mainly been used to store the session keys associated with each entity in the system, and to record the performance metrics transmitted regularly by the MECs. The implementation of BVNF allowed certain vulnerabilities and attacks to be countered. However, the authors did not address other types of attack, in particular the replay attacks.

Bunker has been introduced in (Scheid *et al.*, 2019). so that users can acquire VNFs without relying on a central entity, but by using blockchain properties and smart contracts to address trust concerns regarding the integrity of VNFs packages and VNF package delivery. However, a number of key challenges still remain. No cryptographic method is used to ensure the security of the data in the repository, data storage is not based on a distributed file system, and the prototype does not take into account the VNF solution. In addition, the gas costs associated with the interactions implied in the BUNKER project have not been calculated.

(Kapassa *et al.*, 2020) proposed a smart distributed NFV marketplace design solution based on the principle of a multi-faceted repository, supporting the storage, publication and proposition of VNF/NS developed in a 5G infrastructure, the use of smart contracts have led to an increase in transparency and

trust, as well as verification of license models. However, the model does not include any NFV solution and storage is based on a centralized database rather than a distributed file system.

The contribution of this research resides in the innovative integration of blockchain in Network Functions Virtualisation (NFV) within the 5G world. We focus on the publication and accessibility of NFV models in a framework that engages both network service slice providers and consumers. Our study adopts a decentralized, secure and transparent approach to improve identity management, authorisations associated with NFV models and full traceability of transactions and changes made to NFV configurations. Through the use of smart contracts, we introduce automation into the publishing process, ensuring a flexible and participative governance structure. This not only facilitates the secure and streamlined distribution of NFV models, but also establishes the basis for a trusted environment in which interactions between providers and consumers are immutably recorded.

The paper is structured as follows: the first section introduces explanations of the main concepts used in our contribution, the second section provides an overview of the architecture proposed and implemented to simulate the process, we list the tools and languages used in the third section, the fourth section is dedicated to explaining the methodology, the stages of the contribution and the results, The fifth section provides a discussion and comparison with existing research works. This article ends up with a conclusion.

2. Preliminaries

2.1. Network Slicing and Network Function Virtualisation

Network slicing is one of the key features of 5G, allowing operators to tailor the network and quality of service. A Network Slice is a logical network on a shared infrastructure that provides specific network capabilities and characteristics, supporting various service properties for the customers of the network slicing. Network slices are generally made up of sets of network slice subnets, the network slice subnets are then made up of Virtualized Network Functions (Li *et al.*, 2023).

NFV was introduced as a new concept by ETSI (Dreibholz, 2020), which considers the implementation of 5G Network Functions as software running on the NFV infrastructure (NFVI). NFVI covers the range of physical resources and how these can be virtualized. VNF properties such as resources needed (Compute, Storage, Networking), connectivity (External Connection points, Internal Virtual Links, Internal Connection Points) and lifecycle management behavior (Scaling, Instantiation) are defined within a VNF Descriptor (VNFD). The VNFD provided by the VNF vendor must comply with TOSCA based service template specified in ETSI NFV-SOL001 (Hung *et al.*, 2017). (TOSCA stands for Topology and Orchestration Specification for Cloud Applications and provides a language to describe the topology for cloud services, their components, relationships and the processes for deploying and managing theses services throughout their lifecycle), one of the main parts of the VNFD is VNF topology modeled in a cloud agnostic, containing a set of VDUs (Virtual Deployment Units) that describes the capabilities of virtualized containers such as virtual CPU, RAM, disks...etc (Nguyen *et al.*, 2018).

The VNF package contains the VNFD, the software images used to run the VNF and other optional files containing all the information required for managing the lifecycle of a VNF, the package is delivered by the VNF provider and should be protected from modification and digitally signed, the package is stored in a repository by the NFVO (NFV Orchestration) and can be accessed by VNFM (VNF Management).

In the context of 5G, Network Service Descriptor (NSD) defines the structure of a network service, specifying how different VNFs are connected to create a 5G service within a network slice, it includes details such as quality of service (QoS), latency requirements, bandwidth, security requirements, and other attributes specific to that service (Dreibholz, 2020).

The Network Resource Model (NRM) refers to the abstraction and representation of the underlying physical and virtualized network resources. It is used to manage the resources required to deploy and operate virtualized network functions (VNFs) and network services.

Network Slice Subnet (NSS) - represents the management aspects of a set of managed functions and required resources (e.g. compute, storage and network resources). Its instantiation is called a Network Slice Subnet Instance (NSSI). NSS structure and configuration are described within the Network Slice Subnet Template (NSST).

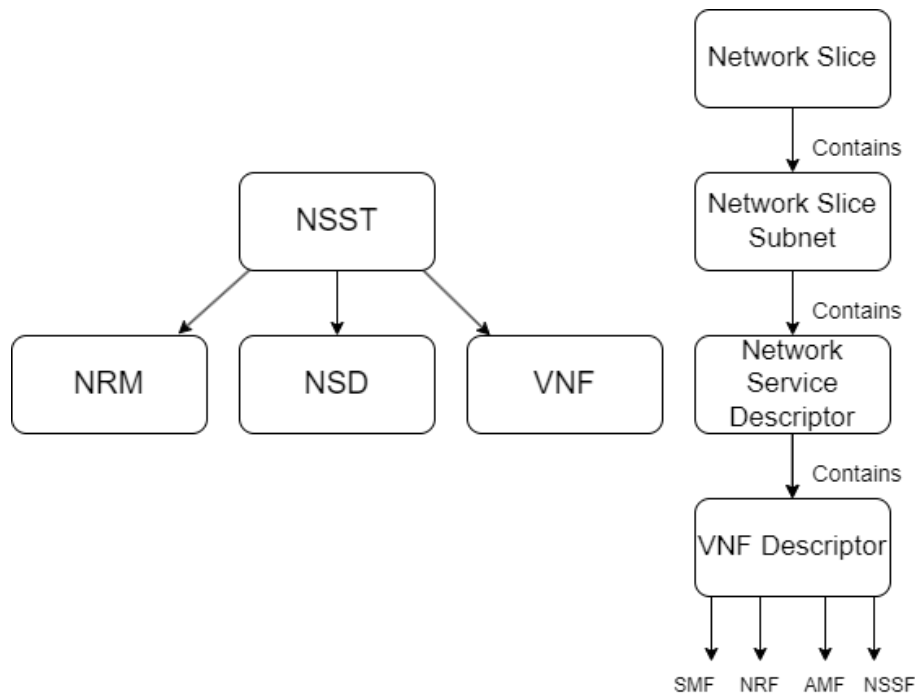


Figure 1. NSST Composition: NRM, NSD, VNF Fusion

2.2. Blockchain

Blockchain is a secure, decentralized technology for storing data or information within an immutable register in a chronological and transparent manner. The technique was first envisaged in 1990 by researchers Haber and stornetta as a decentralized system for time-stamping digital documents and ensuring their integrity (Haber *et al.*, 1991). However, it remained almost unused until it was adapted by Satoshi Nakamoto in 2008, when he (or they) introduced the digital cryptocurrency, Bitcoin. Each block (where a set of transactions are grouped) of the blockchain is linked to the previous and following blocks by means of the hash function, the hash of a block is calculated by considering the block's content and the hash of the previous block, once a block is added to the blockchain, changing any information within it would require recalculating its hash. However, since the hash of the previous block is part of the current block's hash, changing anything in a single block would require changing the entire chain following it. This makes the blockchain tamper-resistant. Blockchain has undergone several evolutions (Mukherjee *et al.*, 2021):

Blockchain 1.0: The first generation of blockchain is blockchain 1.0, which revolutionized the world of payments by removing the need for a central entity to exchange currency, with the introduction of crypto-currencies such as bitcoin.

Blockchain 2.0: Blockchain 2.0 is the extension of blockchain 1.0 for the exchange of the Ethereum cryptocurrency and the introduction of smart contracts, for the design of decentralized applications (DApps).

Blockchain 3.0: Blockchain 3.0 represents an extended approach beyond crypto-currency, based on the use of smart contracts only to create autonomous decentralized organizations.

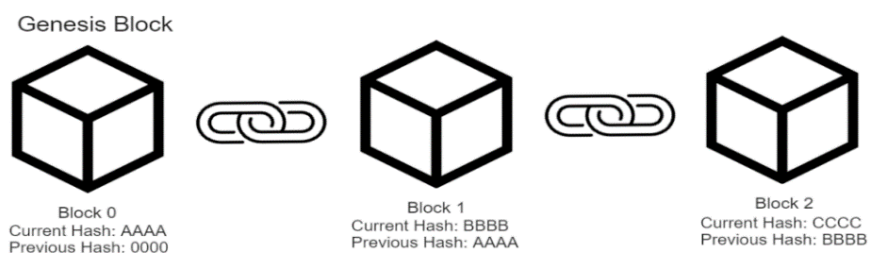


Figure 2. The blockchain structure

2.3. Smart Contracts

Smart contracts are deterministic computer programs using a high-level, Turing-complete programming language, such as Solidity (Taherdoost, 2023). Depending on the terms defined on the contracts, smart contracts provides self-verification and self-execution, and are processed using the Ethereum Virtual Machine (EVM). Contracts are created through transactions, compiled into EVM bytecode and then deployed on the blockchain network, each contract on the blockchain is assigned a unique address and is immutable, modification of a deployed contract is impossible, and any addition to the code results in the creation of a new transaction and the need to incur new costs. In fact, the execution of smart contracts on the ethereum blockchain entails various costs which are measured by means of Gas, the latter serves as a metric indicating the computational effort required to process and execute smart contract transactions or operations. Gas fees are settled in Ether (ETH), the native crypto-currency of the Ethereum network, and expenses associated with deploying and launching a smart contract on Ethereum could vary depending on the complexity of the contract (Bistarelli *et al.*, 2020).

2.4. IPFS

IPFS is a peer-2-peer (P2P) distributed file system based on a distributed hash table (DHT). The system allows any user on the network to host and receive content via their address. When storing data, it divides files into smaller chunks that are cryptographically hashed using the sha2-256 algorithm, which generates a unique 256-bit or 32-byte long fingerprint called a content identifier (CID), a self-describing content address identifier that forms an address based on the content itself. When a user requests a specific chunk, the request is passed through the DHT to the nodes hosting the corresponding hash. All the required chunks are then merged to build the main object and present it to the user (Benet, 2014).

2.5. 5G NFV and network slicing security challenges

The adoption of 5G networks is driving the use of Network Function Virtualisation (NFV) technologies. NFV, like many emerging technologies, poses new security challenges which, in turn, increase existing security issues for NFV and 5G networks: for example a threat vendor could obtain privileges to embed a malicious network slice containing malicious VNFs. It is therefore imperative to systematically address the security issues and focus on implementing robust security mechanisms in 5G NFV systems, by taking into account certain points (Madi *et al.*, 2021):

- Each modification made to the VNF package must be subject to version control and monitoring.
- The NFV platform must support redundancy and backup mechanisms to improve its overall reliability.
- Use access control management to ensure that only authorized entities have access to specific management functions.
- Ensure secure delivery of the Virtual Network Function (VNF) package from the NFV provider to the VNF consumer (telecoms operator). This involves using secure channels, validating the integrity of the VNF packages and digitally signing them using the VNF provider's private key. The certification authority issues digital certificates which are used to authenticate the parties involved in the communication.

3. Proposed System Architecture

In this paper, we propose a new framework that enhances the previously specified requirements using Blockchain 2.0 technology aspects. The transparent and immutable nature of blockchain ensures a tamper-resistant audit trail for monitoring changes, hence, we use blockchain for version control by recording each modification to the VNF package as a new transaction, Smart contracts can enforce rules by allowing specific roles to execute predefined functions within the NFV platform, such as updating versions, ensuring that only authorized parties can make modifications to the NFV operational aspects. Eliminating certificates issued by a central authority in favor of a blockchain-based approach can offer advantages in terms of decentralization and autonomy; the VNF provider can sign the VNF package directly using the private key issued from its account address. Customers have the opportunity to establish a trusted relationship with the provider through registration using their account address.

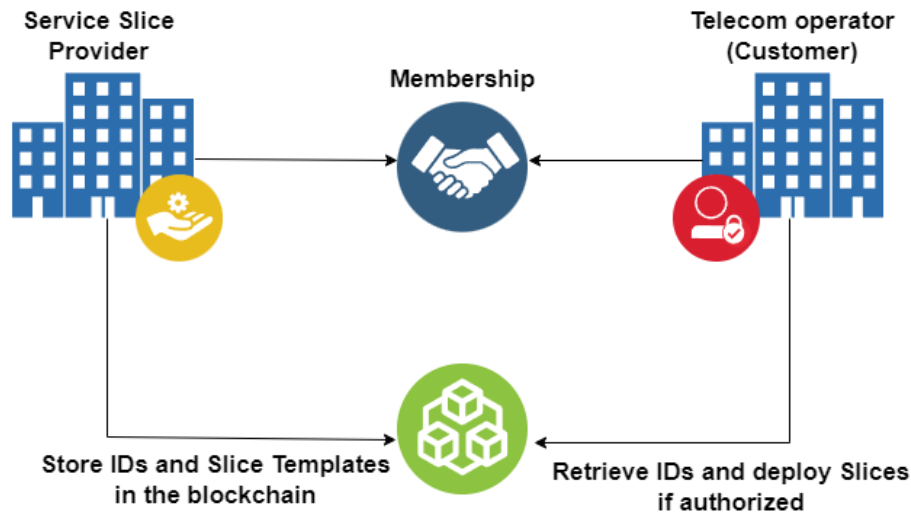


Figure 3. Logical architecture of proposed work

4. Experimental Setup

To implement our framework, we used free5gmano, a project that involves the development of a 5G MANO (Management and Network Orchestration) system based on 3GPP TS 28.531 and TS 28.532 Release 15 (R15) standards. Its objective is to facilitate the management and scheduling of 5G network slices, this project is built on kube5gnfvo as a platform for deploying network slice subnet instances (NSSIs) for free5gc's VNFs functions. Smart contracts are written in the Solidity language and deployed on a local ethereum blockchain emulator, ganache-cli, which generates a number of accounts at startup with their associated private keys and balances. We use the web3.py Python library to interact with Ethereum, which allows us to deploy smart contracts, send transactions, interact with the deployed contracts and read data from the blockchain using python code. We keep a backup copy of VNF packages and all their modifications in IPFS (Interplanetary File System), which we interact with using the py-ipfs-http-client library.

5. Contribution methodology and results

This research represents a new contribution to improving the security of Virtual Network Function packages and the Network Service Slicing instantiation in a 5G network environment by including a blockchain-based distributed network between the network service slice provider and customers, typically the telecom operators. Our approach uses smart contracts to store various identifiers such as VNFs, NRMs and NSDs. These identifiers are verified on the blockchain before being combined to form an NSST. Customers are registered by the service provider using smart contracts, establishing a relationship of trust. This registration gives customers exclusive access, allowing them to initiate NSSI allocations using the allocated NSST IDs. The proposed framework ensures that only registered clients can list available NSS templates, and introduces nonces to prevent replay attacks during NSSI allocation requests.

We have implemented three different smart contracts: one for customer registration and authentication, a second for storing the various identifiers generated during template creation, enabling the accuracy of a specific identifier to be verified in the blockchain and the appropriate identifier to be provided to an eligible customer. The third contract facilitates the processing of requests for the allocation of network service slice instances; it is also responsible for generating nonces and ensuring the freshness of each request during verification. The three contracts are deployed by the service provider, which therefore owns them.

5.1. Smart Contract Description

register.sol: The contract specifies an "admin" address which is initialized by the address of the account that deployed the contract (Network Service Slice Provider), and the account only that has the right to register customers (Telecom Operators). The contract maintains an up-to-date list of registered customer addresses, helping to avoid duplicate registrations and to verify the registration status.

Contract 1: register

```

1: array: customerRegistered
2: constructor admin ← msg.sender
3: function customerRegistration(_address)
4:   require admin == msg.sender
5:   for each customer in customerRegistered Do
6:     if customerRegistered [ customer ] == _address Then
7:       revert ("Customer Already Registered")
8:     end if
9:   end for
10:  push customer in customerRegistered
11: end function
12: function isCustomerRegistered (_address)
13:   for each customer in userRegistered Do
14:     if customerRegistered [customer] == _address Then
15:       return true
16:     else
17:       return false
18:     end if
19:   end for
20: end function

```

store.sol: The contract is deployed for storing and managing different template identifiers, the storage process is exclusively controlled by the contract administrator. Furthermore, the contract features functions to verify the existence of a template based on its unique identifier and retrieve the template identifier associated with a specified type.

Contract 2: store

```

1: struct template (templateID, type)
2: array templates
3: constructor admin ← msg.sender
4: function SetTemplateID (newTemplateID, newType)
5:   require admin == msg.sender
6:   newTemplate = template (newTemplateID, newType)
7:   push newTemplate in templates
8: end function
9: function templateExists (templateIdToCheck)
10:  for each template in templates Do
11:    if keccak256(abi.encodePacked(template.templateId)) ==
        keccak256(abi.encodePacked(templateIdToCheck)) Then
12:      return true
13:    else
14:      return false
15:    end for
16: end function
17: function getTamplateIDbyType ( )
18:  for each template in templates Do
19:    if keccak256(abi.encodePacked(template.Type)) ==
        keccak256(abi.encodePacked("nsst")) Then
20:      return template.templateId
21:    end if
22:  end for
23: end function

```

Allocate.sol: The contract uses a blockchain-based method to enable customers to generate a nonce by extracting it from the hash of the previous block. The "addRequest" function facilitates the inclusion of different customer requests in the ledger. In particular, the contract offers a "verifyRequest" function that allows the service provider to verify the authenticity of a request by comparing the nonce and customer account identifier provided with those recorded in the ledger.

Contract 3: allocate

```

1: struct (nonce, accountID)
2: array requests
3: function generateNonce ( ) returns uint256
4:   bytes32 blockHash ← blockhash(block.number - 1)
5:   uint256 blockHashUint ← uint256(blockHash)
6:   uint256 nonce ← blockHashUint & 0xFFFFFFFFFFFFFFFF
7:   return nonce
8: end function
9: function AddRequest ( _nonce, _accountID)
10:  request newRequest ← request(_ nonce, _accountID)
11:  push request in requests
12: end function
13: function verifyRequest ( _nonce, _accountID)
14:  for each request in requests Do
15:    if request.nonce == _nonce AND request.accountId == _accountID Then
16:      return true
17:    else
18:      return false
19:    end if
20:  end for
21: end function

```

5.2. Contribution steps:

The contribution consists of several phases, as illustrated below and detailed further on.

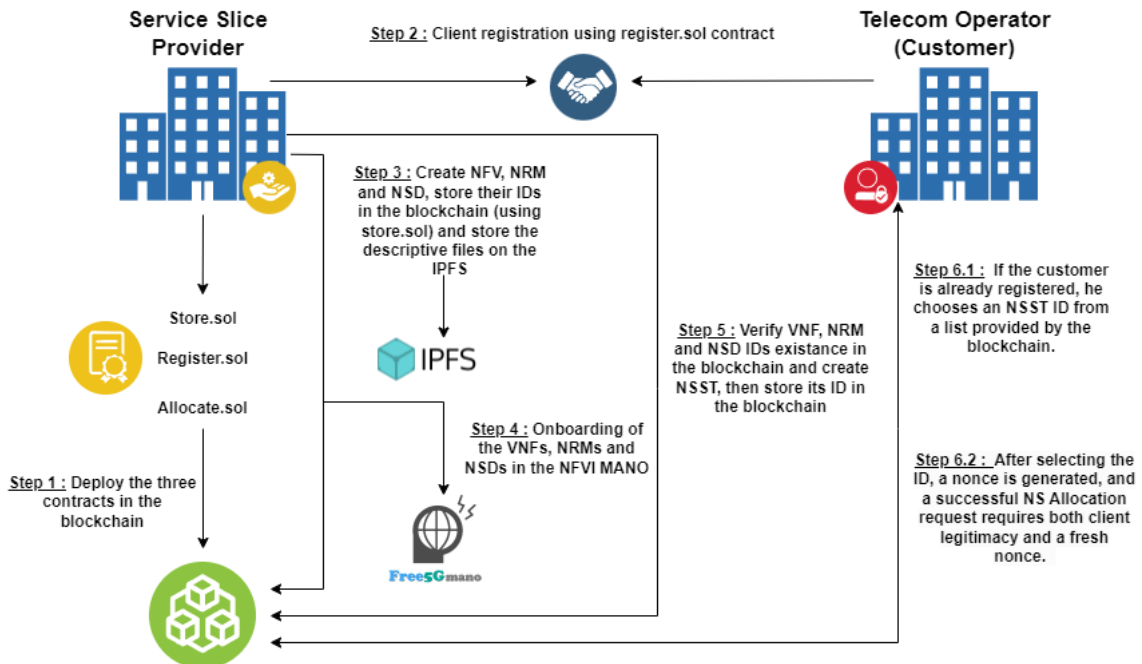


Figure 4. Different steps of our framework

Phase 1 – Contract deployment: The network service slice provider starts by deploying smart contracts required for customer registration, templates creation and storage, and slice instantiation on the blockchain network and publishing their addresses to future customers.

Phase 2 - Client Registration Phase: Customers of the same blockchain network, each with an address, register with the service provider using the "customerRegistration" function that belongs to the "register.sol" contract and is only invoked by the service provider (the contract owner). This registration is essential for checking that customers are eligible to instantiate network subnet slices.

Phase 3 - VNF package management phase: In parallel to the second phase, the service provider manages VNF packages containing all the elements required to deploy and run a set of virtualised network functions (VNF such as 5G Access and mobility function) via the Kubernetes orchestrator on a virtualised network function infrastructure (VNFI). This includes the configurations described in the VNFD (ETSI GS NFV-SOL 001 compliant and structured according to TOSCA standards). In addition, the service provider creates the Network Resource Model (NRM), which provides a unified view of available physical and virtual resources, and the Network Service Descriptor (NSD), which details specific network services.

The service provider creates the various models and registers their IDs in the blockchain by calling the function "setTemplateID", it also registers the various descriptive files on the IPFS network. These registrations and creations are only performed by the service provider using the various functions of the store.sol contract, which is under its ownership.

Phase 4 – Onboarding Phase: Integration of VNFs, NRMs and NSDs into the NFVI (Network Function Virtualization Infrastructure) MANO (Management and Orchestration) system involves registering them in the catalog, a critical step before proceeding to the next phase (5).

Phase 5 – Network Service Subnet Template Creation Phase: To create an NSST, the service provider populates three IDs, initiating an ID verification process to ensure their legitimacy within the blockchain using the function "templateExists". Upon successful creation of the NSST (Network Service Subnet Template), its ID is stored in the blockchain before being published in the catalog.

Verification, creation of NSST and storage are carried out exclusively by the service provider using the functions of the "store.sol" contract of which he is the owner.

Phase 6 – Network Slice Subnet Instantiation: When telecom operators, acting as clients, seek to deploy network slices with 5G network functions, they request the service provider to list available NSST IDs by calling the function "getTemplateIDbyType". To obtain an ID, the service provider must authenticate the client and verify their permissions using "isCustomerRegistered" function. Upon approval, available NSST IDs are returned.

The client selects an NSST ID appropriate to the deployment requirements and submits a Network Slice Subnet Instance request. To do this, it first generates a nonce (number used once) with its address being verified in the blockchain. Once the nonce has been generated, it sends the request, indicating its identifier. The service provider checks the freshness of the nonce to avoid any malicious duplication, verifies the client's identifier and authorises instantiation. The request received is then recorded in the blockchain for subsequent verification.

5.3. Computational effort measurement

The term "gas" refers to the unit of measurement of the computational effort needed to execute operations or transactions. Each transaction in a smart contract consumes a set amount of gas, and the associated cost of gas for each transaction is determined by various factors, such as the complexity of the transaction and current network conditions. The following table gives the values of each transaction cost (gas cost) made during our six phases:

Table 1. Gas cost associated to each operation

Operation	Gas cost
Deployment of the store.sol contract	286156
Deployment of the register.sol contract	694620
Deployment of the allocate.sol contract	262406
"setTemplateID" call function : VNF	127821
"setTemplateID" call function : NRM	112821
"setTemplateID" call function : NSD	112821
Register a client	65163
Create NSST	112833
Instantiate NSSI	83822

The “getTemplateIDbyType”, “isCustomerRegistered” and “templateExists” functions do not generate any gas costs as they are "view" type functions. In the Solidity programming language, a function whose only objective is to read the state variables defined in the contract and not to modify them is referred to as a “view function”.

6. Discussion

This research work investigates the major challenges associated with the risk of intrusions and security breaches when deploying network slicing and NSST templates. Our proposed framework focuses on securing the authenticity and versioning of VNF, NRM, NSD and NSST templates by ensuring their integrity and registering them with unique identifiers. Access to modify or add new templates is subjected to strict verification to ensure that only the network service slice provider have the required authorisations. In addition, we have established a trusted relationship between customers and the provider: each customer wishing to acquire a network slice sub-network instance must be a member of the blockchain network and register with the provider to obtain the necessary authorisations. The submission of a request by a registered client takes the form of a transaction signed with its private key before being sent. The Network Service Slice Provider then verifies the legitimacy of the client by ensuring that it is registered before processing the request and providing it with the instance, thereby avoiding any risk of a man-in-the-middle attack. In addition, our framework integrates a virtualised network function orchestration management platform for 5G networks, an aspect not addressed by works (Mishra *et al.*, 2020), (Scheid *et al.*, 2019) and (Kapassa *et al.*, 2020) cited in the bibliography. We have also introduced the use of a randomly generated nonce, which is only accessible to clients registered with the provider. This nonce, calculated from the hash of the previous block of the blockchain, is included in the request for a network slice subnet instance. The provider checks the uniqueness of the nonce; if the nonce has been previously used, it can deduce that a previous request has been copied by a malicious actor and reject the request. This approach enhances security by preventing replay attacks, a consideration absent from research articles (Mishra *et al.*, 2020), (Scheid *et al.*, 2019) and (Kapassa *et al.*, 2020). Table 2 provides a comparative analysis of our work with other related works. The table uses three indicators: "Yes" to indicate availability, "No" to indicate the lack of availability and “-“ to indicate not sufficient information.

Table 2. Comparative analysis of our work and related works

Characteristic	B-VNF (Mishra <i>et al.</i> , 2020)	BUNKER (Scheid <i>et al.</i> , 2019)	(Kapassa <i>et al.</i> , 2020)	Our Work
Cost of various Smart Contracts	Yes	No	No	Yes
Integration of VNF orchestration platform	No	No	No	Yes
VNF package storage	Yes	Yes	No	Yes
Authentication	Yes	Yes	-	Yes
Eliminates the risk of replay attacks	No	No	No	Yes
Based on decentralized storage	Yes	Yes	No	Yes

7. Conclusions

Our research work presents a comprehensive blockchain-based security framework designed to improve the integrity and exclusivity of virtual network function (VNF) packages within network slicing services (NSS). By taking advantage of smart contracts, we have built a secure system for customer registration, identity verification and NSSI allocation. This framework not only guarantees the reliability of customer interactions, but also mitigates the risk of replay attacks through the introduction of random numbers, called nonces generated from previous block hashes. Our contribution represents a significant advance in network security, providing a reliable solution for protecting VNF packages in 5G environments. We have outlined the major advances resulting from the comparison of our research with the literature, highlighting the marked improvements achieved. Through further validation and implementation, our framework can support increased trust and security in the telecommunications sector, paving the way for more resilient and robust network infrastructures.

References

1. Alemany, P., Vilalta, R., Muñoz, R., Casellas, R., R. Marínez. (2020) Peer-to-Peer Blockchain-based NFV Service Platform for End-to-End Network Slice Orchestration Across Multiple NFVI Domains. In: *2020 IEEE 3rd 5G World Forum (5GWF)*, 2020, 151-156. Bangalore, India: IEEE.
2. Attaoui, W., Sabir, E., Elbiaze, H., Guizani, M. (2023) VNF and CNF Placement in 5G: Recent Advances and Future Trends. *IEEE Transactions on Network and Service Management*, 20(4), 4698-4733. DOI: 10.1109/TNSM.2023.3264005.
3. Bala, R., Manoharan, R. (2022) Blockchain based Secure and Effective Authentication Mechanism for 5G Networks. In: *2022 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)*, 2022, 1-6. Pune, India: IEEE.
4. Benet, J. (2014) IPFS - Content Addressed, Versioned, P2P File System, *ArXiv*, 1407.3561.
5. Bistarelli, S., Mazzante, G., Micheletti, M., Mostarda, L., Sestili, D., Tiezzi, F. (2020) Ethereum smart contracts: Analysis and statistics of their source code and opcodes. *Internet of Things*, 11, 2542-6605. DOI: 10.1016/j.iot.2020.100198.
6. Dreiholz, T. (2020) A 4G/5G Packet Core as VNF with Open Source MANO and OpenAirInterface. In: *2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2020, 1-3. Split, Croatia: IEEE.
7. Haber, S., Stornetta, W.S. (1991) How to Time-Stamp a Digital Document. *Journal of Cryptography*, 3, 99-111. DOI: 0.1007/3-540-38424-3_32.
8. Hung, Y. -M., Chien, S. -C., Hsu, Y. (2017) Orchestration of NFV virtual applications based on TOSCA data models. In: *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2017, 219-222. Seoul, Korea (South): IEEE.
9. Kapassa, E., Touloupou, M., Kyriazis, D., Themistocleous, M. (2020) A Smart Distributed Marketplace. In: *Themistocleous, M., Papadaki, M. (eds) Information Systems. EMCIS 2019*, Springer.
10. Li, H and Kong, Zi and Chen, Y and Wang, L and Lu, Z and Wen, X and Jing, W and Xiang, W. (2023) Slice-Based Service Function Chain Embedding for End-to-End Network Slice Deployment. *IEEE Transactions on Network and Service Management*, 20(3), 3652-3672. DOI: 10.1109/TNSM.2023.3250719.
11. Liu, J and Huang, C. -T. (2021) Efficient and Trustworthy Authentication in 5G Networks Based on Blockchain. In: *2021 International Conference on Computer Communications and Networks (ICCCN)*, 2021, 1-6. Athens, Greece: IEEE.
12. Madi, T., Alameddine, H -A., Pourzandi, M., Boukhtouta, A. (2021) NFV security survey in 5G networks: A three-dimensional threat taxonomy. *Computer Networks*, 197. DOI: 10.1016/j.comnet.2021.108288.
13. Mishra, R., Kalla, A., Shukla, K., Nag, A., Liyanage, M. (2020) B-VNF: Blockchain-enhanced Architecture for VNF Orchestration in MEC-5G Networks. In: *2020 IEEE 3rd 5G World Forum (5GWF)*, 2020, 229-234. Bangalore, India: IEEE.
14. Mukherjee, P., Chittaranjan, P. (2021) Blockchain 1.0 to Blockchain 4.0—The Evolutionary Transformation of Blockchain Technology. *Blockchain Technology: Applications and Challenges*, 203, 29-49. DOI: 10.1007/978-3-030-69395-4_3.
15. Nguyen, T. -H., Yoo, M. (2018) A VNF Descriptor Generator for Tacker-based NFV Management and Orchestration. In: *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, 2018, 260-262. Jeju, Korea (South): IEEE.
16. Patel, W., Tripathy, P. (2023) eMBB, URLLC, and mMTC 5G Wireless Network Slicing: A Communication-Theoretic View. In: *2023 International Conference on Artificial Intelligence and Smart Communication*, 2023, 1291-1296. Greater Noida, India: IEEE.
17. Scheid, E.J., Keller, M., Franco, M.F., Stiller, B. (2019) BUNKER: A Blockchain-based trUsted VNF pacKagE Repository. In: *International Conference on the Economics of Grids, Clouds, Systems, and Services GECON 2019*, 188–196. Springer.
18. Taherdoost, H. (2023) Smart Contracts in Blockchain Technology: A Critical Review, *Information*, 14(2). DOI: 10.3390/info14020117.
19. Zhang, Y., Deng, R. H., Bertino, E., Zheng, D. (2021) Robust and Universal Seamless Handover Authentication in 5G HetNets. *IEEE Transactions on Dependable and Secure Computing*, 18(2), 858-874. DOI: 10.1109/TDSC.2019.2927664.