Blockchain-Based Secure Authentication and Authorization Framework for Robust 5G Network Slicing

Shalitha Wijethilaka, *Student Member, IEEE*, Awaneesh Kumar Yadav, *Member, IEEE*, An Braeken, *Senior Member, IEEE*, and Madhusanka Liyanage, *Senior Member, IEEE*

***Abstract*—The rapid evolution of heterogeneous applications signifies the requirement for network slicing to cater to diverse network requirements. Network Functions (NFs), which are the essential elements of network slices, are required to communicate with each other securely to facilitate network services. Certificates are the established method to authenticate each other. However, dynamic certificate management while allowing NFs to communicate in a multi-operator environment is arduous. Also, sharing NFs between network slices originates authorization-related security challenges such as unauthorized service utilization, deceptive Denial of Service attacks, and data leakages from network slices. In this paper, we develop a novel framework to address the security challenges related to authentication and authorization in 5G network slicing systems. A blockchain-based multi-party distributed certificate management framework with secure communication protocols is developed using elliptic curve cryptography to facilitate certificate services for multi-operator environments. Also, we propose a blockchain-based NF authorization framework to mitigate the security vulnerabilities in NF sharing between network slices. We implement the proposed framework using Hyperledger Fabric blockchain with Java chain codes and perform comprehensive experiments to show the significance of our framework.The Ability to mitigate the single point of failure with respect to state-of-the-art, including traditional certificate authorities and blockchain-based certificate authorities, time analysis for certificate generation, and the potential to eliminate the mentioned authorization attacks are some of the experiments conducted.Also, we have shown that our framework is secure using informal and formal (using Real-Or-Random (ROR) logic and Scyther Validation tool) security verification mechanisms.**

***Index Terms*—5G, network slicing, blockchain, authentication, authorization, security, certificate.**

Manuscript received 8 November 2023; revised 8 April 2024; accepted 2 June 2024. Date of publication 19 June 2024; date of current version 21 August 2024. This work is partly supported by COST Action CA22104 Beingwise, Cybersecurity Research Program Flanders - second cycle (VOEWICS02), and the Science Foundation Ireland under CONNECT phase 2 (Grant no. 13/RC/2077_P2) project. The associate editor coordinating the review of this article and approving it for publication was M. F. Zhani. *(Corresponding author: Shalitha Wijethilaka.)*

Shalitha Wijethilaka and Awaneesh Kumar Yadav are with the School of Computer Science, University College Dublin, Dublin 4, D04 V1W8, Ireland (e-mail: mahadurage.wijethilaka@ucdconnect.ie; awaneesh.yadav@ucd.ie).

An Braeken is with the Department of Engineering, Technology (INDI), Vrije Universiteit Brussel, 1050 Ixelles, Belgium (e-mail: an.braeken@vub.be).

Madhusanka Liyanage is with the School of Computer Science, University College Dublin, Dublin 4, Ireland, and also with the Centre for Wireless Communications, University of Oulu, 90570 Oulu, Finland (e-mail: madhusanka@ucd.ie).

Digital Object Identifier 10.1109/TNSM.2024.3416418

I. Introduction

THE RAPID evolution of the Internet and related technologies causes significant advancements in heterogeneous applications such as smart healthcare, autonomous vehicles, Internet of Things (IoT), and industrial automation [1]. For instance, Lian highlighted that the number of IoT devices will exceed 25 billion by 2025 [2]. Connectivity is one of the fundamental requirements for the realization of these diverse applications. Also, the connectivity requirements are diverse in these applications. Therefore, future telecommunication networks, including Fifth Generation (5G), are being designed to facilitate the network requirements of these diverse applications.

Network slicing, one of the predominant technologies in future telecommunication networks, can divide the physical network into multiple logical networks specific to different applications [3]. In [4], Wijethilaka et al. showed the significance of network slicing for the realization of different applications. Allocating different network slices for different applications allows us to facilitate the diverse communication requirements of respective applications. According to ResearchDive, there is exponential growth in the network-slicing market in different applications. For instance, they expect the healthcare market in network slicing will be $182.5 million by 2027, which is $36.1 million in 2019 [5]. Security is a critical aspect of a network-slicing ecosystem. In addition to the conventional security vulnerabilities in telecommunication systems, network slicing itself introduces a novel threat space. In [6], Olimit et al. identify the potential threat space of network slicing in 5G networks. These security vulnerabilities affect all the users in the network.

The functionality of a particular network slice depends on the deployed Network Functions (NFs). Network Function Virtualization (NFV) allows us to transform NFs from hardware-based entities to software components that can be deployed in commodity servers [7]. Virtual NFs (VNFs) are specific instances of a network function that have been virtualized and deployed as software applications. VNFs provide the building blocks of NFV deployments. According to the Third Generation Partnership Project (3GPP), these VNFs use certificates to authenticate with each other and to enable secure communication [8]. However, certificate management in an operator environment is an unsolved issue yet [9].

As concepts such as virtual network operators, Local 5G Operators (L5GOs) or private 5G Operators are evolving in future telecommunication networks, network slices need to be spanned over multiple administrative domains. Therefore, inter-operator visibility for the certificates is required to ensure secure communication. Standalone deployment of a Certificate Authority (CA) in an operator environment can not provide the required visibility for the certificates over multiple administrative domains. Also, acquiring certificates from commercial CAs is not feasible due to the dynamic nature of NF deployment and the cost of commercial CAs. Therefore, a novel authentication framework is required to manoeuvre the certificate management process related to NFs in network slices of operators.

NFs can be shared between multiple slices or deployed within a single slice. Sharing of NFs between network slices originates a set of authorization security challenges. The company Adaptive Mobile Security (AMS) identified potential security vulnerabilities in NF sharing between network slices [10]. Data leakage between network slices, Denial of Service (DoS) attacks against a particular slice, and unauthorized resource usage are some identified security challenges in NF sharing. Since sensitive data, such as health data, are transmitted over network slices, information leakage creates a serious issue. Also, continuous communication without disruption is crucial for applications such as autonomous vehicles. Moreover, network slice owners might need to pay for unutilized services if some other parties utilize services impersonating them. The network-slicing ecosystem should be strengthened to mitigate these discussed authentication and authorization challenges to ensure the optimal utilization of network slicing.

Blockchain is a novel technology that builds trust between multiple parties using distributed and immutable ledger technologies [11]. The properties, such as accountability, immutability, and increased transparency, motivate us to utilize blockchain in other applications in addition to crypto. Thus, several blockchain-based Public Key Infrastructures (PKIs) have been proposed in the literature. However, almost all of them are based on a central CA, and they use the blockchain as an intermediate technology to store and revoke certificates.

The central CA can be a single point of failure and can also affect the whole ecosystem if the CA is malicious [12]. Therefore, a distributed certificate generation mechanism is required to reduce the impact of malicious individual CAs. Moreover, blockchain, along with its processing power achieved through smart contracts, can be utilized as a supporting technology in our framework as it allows us to make the certificate generation process more trusted, accountable, and transparent. In addition, blockchain technology permits simplifying certificate validation and revocation procedures. Hence, our framework can perform all the certificate operations, including certificate generation, validation, and revocation. Moreover, the same blockchain can be used to eliminate unresolved authorization issues in the slicing ecosystem.

Therefore, we present a blockchain-based framework to handle certificates in a slicing ecosystem and to mitigate authorization vulnerabilities highlighted in this paper. Our framework eliminates the single point of failure in the certificate issuance process. Also, trust is distributed as we utilize multiple parties to issue a single certificate. Therefore, we address the CA trust issues. Lightweight certificate utilization in our framework increases storage and bandwidth efficiency. Since we utilize the blockchain to handle certificates, certificate revocation can be announced to all parties effectively. The secure protocols to receive certificates presented in this paper allow us to deploy our framework with any blockchain network, either public or private. Moreover, our framework mitigates the challenge of NF authorization in network slices.

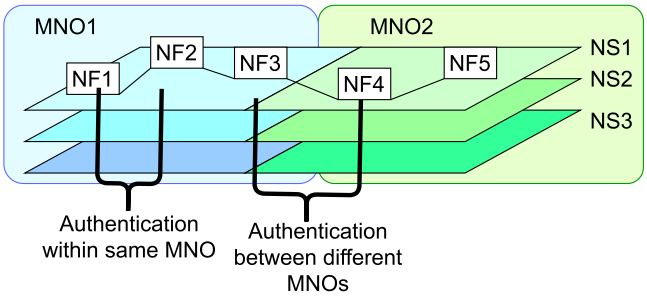

Fig. 1. Mutual authentication of network functions.

A. Motivation

This subsection describes the scenarios that motivated us to conduct this research.

1) Authentication Challenges for Network Functions in Network Slices: A network slice consists of multiple network functions, and they need to communicate with each other to realize the functionality of the network slice. Moreover, network functions need to communicate to realize the multi-domain network slicing, as shown in Figure 1. Even though 3GPP specifies the use of digital certificates for the mutual authentication of network functions, the specific way to manage these certificates is not investigated. If a commercial CA is used, the cost would be very high for the MNO. If a private CA is used, the certificates only will be visible to the internal environment. Therefore, authentication among different MNOs will not be possible. Considering these limitations, a novel certificate management framework is required to enable certificate visibility among different MNOs and reduce the cost of certificates.

2) Security Vulnerabilities in Network Slicing Due to Network Function Sharing Between Slices: Even though network slicing supports diverse application realization, it introduces a new threat space to its users. In [13], Cunha et al. discuss the leading security challenges, such as impersonation attacks, end device vulnerabilities, and compromised NFs, in the packet core due to the network slicing utilization. The authors in [6] highlight potential security challenges and suggest possible solutions. AMS discovers a set of security concerns in 5G network slicing Service Based Architecture (SBA) [10]. In all these research works vulnerabilities in NFs have been highlighted. NF sharing between multiple network slices introduces a set of authorization security challenges. As an industry leader in telecom security, AMS accentuates three security issues in NF sharing between network slices that are

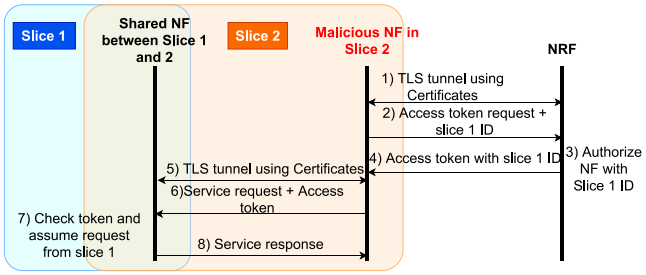

Fig. 2. Unauthorized network service utilization impersonating a legitimate entity.

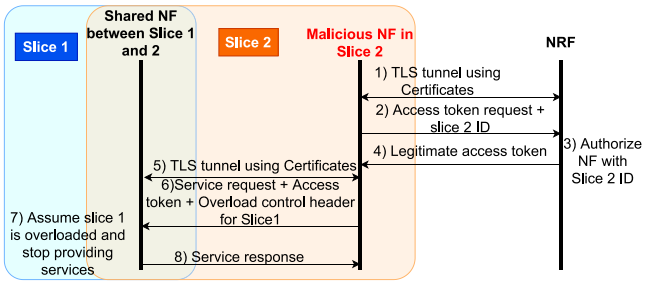

Fig. 3. Service disruptions in a network slice.

not covered in existing standardization [10]. Therefore, this paper aims to provide a solution to the following identified issues, as these issues greatly impact different applications.

a) Unauthorized service utilization: Figure 2 shows the attack flow related to this attack. Here, the Network Repository Function (NRF) acts as a centralized repository for all the 5G NFs in the operator's network. When a particular network function is shared between two network slices, a malicious network function can send requests with slice identities that do not belong to that particular network function but belong to the shared NF. However, the shared NF can not differentiate whether the received request is legitimate or not. Therefore, the actual slice owners may have to pay for the service that they have not used. This is a significant challenge for network slice owners. Due to the massive number of different applications, a massive number of network slices are available for different applications. Therefore, the impact of this attack on network slice owners is not negligible.

b) Deceptive denial of service attack on network slices: In 5G, an overload control header is used to indicate that a particular slice is overloaded. Figure 3 shows the flow diagram related to attack 2. As mentioned in the earlier attack, the malicious NF can send the overload control headers with some other slice's identity to the shared NF. Then, the shared NF assumes that the slice with the received ID is overloaded, and it stops providing services to the particular slice. That means the malicious NF can indirectly perform a DoS attack on some other slice due to the lack of solid authorization mechanisms. This is a severe issue for network slices. Critical application domains, such as autonomous vehicles and military applications, would be endangered due to this vulnerability.

c) Data leakage between slices: As shown in Figure 4, the malicious NF can receive authorizations of slices that actually do not belong to them. Then, the malicious NF can request the information of data that flows across the unauthorized

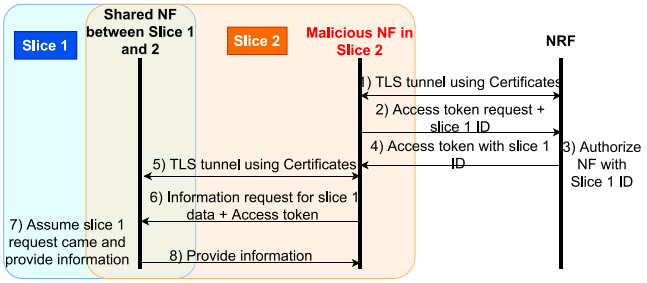

Fig. 4. Accessing data in another slice causing privacy challenges.

slice from the shared NF using the malicious authorization. This vulnerability can cause data leakage between network slices. Sensitive application domains such as smart healthcare and smart grids are subjected to data privacy and security challenges due to this vulnerability.

B. Our Contribution

The discussed limitations in the network-slicing ecosystem motivate us to propose a blockchain-based framework to address authentication and authorization-related security challenges. The contribution of this paper can be enumerated as follows.

- Propose a novel multi-party fully distributed PKI framework to handle certificates in an operator environment to mitigate authentication challenges. To the best of our knowledge, this is the very first fully distributed blockchain-based PKI framework that does not require the service via root CA. The distributed nature can eliminate centralized CA owners' unfair controlling/influence issues and the single point of failure issues of traditional root CA systems.
- Formulate the protocols required for the certificate generation process so that it can be implemented in any blockchain network.
- Propose a blockchain-based authorization framework to mitigate security attacks (i.e., unauthorized service utilization, deceptive DoS attacks, and data leakages from network slices) in NF sharing between slices. This work pioneers mitigating these vulnerabilities in the slicing environment.
- Conduct an extensive formal (using ROR logic and Scyther tool) and informal verification to show the security of Mobile Network Operator (MNO) certificate issuance and NF certificate issuance procedures.
- Implement the proposed framework in a testbed by using existing blockchain technology (Hyperledger Fabric) to demonstrate the proposal's feasibility and evaluate the proposed framework's performance advantage over state-of-the-art systems.
- Practically implement the NS-related security attacks (i.e., single point of failure of the centralized CA in a slicing ecosystem, unauthorized service utilization, deceptive DoS attacks, and data leakages from network slices) and verify the capability of the proposed framework to mitigate the identified attacks.

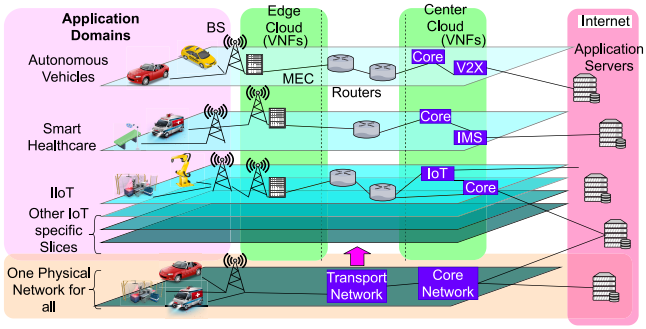

Fig. 5. Transformation of traditional telecom networks to network slicing system to support different applications.

C. Paper Outline

The paper consists of nine sections. Section II presents the related background of this paper, including network slicing, Elliptic Curve Qu-Vanstone (ECQV) certificates, security vulnerabilities in NF sharing, and related works, along with a comprehensive comparison. Section III presents the proposed framework for certificate management and NF authorization with the threat model and security features. Section IV provides comprehensive experiments performed in a real blockchain-based implementation using Hyperledger Fabric. An extensive formal and informal security analysis of the protocols of our framework is presented in Sections V and VI, respectively. Section VII highlights the limitations of our framework and potential solutions for them. Finally, Section VIII concludes the paper with potential future research directions.

II. Background

In this section, we will provide the required background related to this paper. The significance of network slicing, ECQV certificates, security vulnerabilities in NF sharing, and existing research related to our paper are described here.

A. Network Slicing

The expansion of diverse applications such as autonomous vehicles, the military, and healthcare intensifies heterogeneous communication requirements. For instance, real-time, ultra-reliable connectivity is required for autonomous vehicles, but for environmental monitoring applications, those are not critical. Traditional telecommunication networks can not facilitate these diverse requirements using a single physical network. Deploying a physical network for each application is also not feasible due to the cost. Network slicing has the potential to address this problem in different application domains. Figure 5 shows how traditional telecommunication networks are evolving to a network-slicing-enabled environment to realize diverse applications.

B. Elliptic Curve Qu-Vanstone (ECQV) Implicit Certificates

Generally, public-private key schemes such as Rivest–Shamir–Adleman (RSA) are computationally expensive. However, Elliptic Curves (ECs) allow us to develop public-private key schemes which can achieve higher security levels

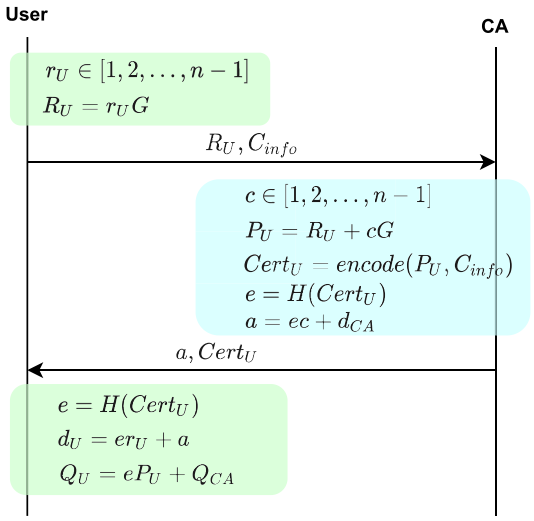

Fig. 6. The flow diagram of ECQV certificate generation.

with smaller key lengths [14]. An elliptic curve in a finite field F_p can be denoted as $y^2 = x^3 + ax + b$, where a, and b are constants in F_p such that $\Delta = 4a^3 + 27b^2 \neq 0$. We denote the generator of the curve by G.

Digital certificates are the best-known method for establishing digital identities in network communications. However, traditional certificates, also known as explicit certificates, have several limitations in terms of infrastructure, memory, and bandwidth. Also, acquiring a publicly trustworthy certificate is a very costly operation (around $60 per year on average) [15]. Implicit certificates enable a low-resource trust model where the public key and the digital signature of a particular certificate are superimposed together so that the required bandwidth is minimal, as the certificate and the verification key are not required to transmit together. Implicit certificates are faster and smaller than conventional explicit certificates.

ECQV certificates are a kind of implicit certificate that is designed to perform the functionality of the general certificates in an environment where resources, such as bandwidth, computing power, and storage, are limited [16]. Figure 6 shows the process of generating an ECQV certificate. Initially, the user (U) selects a random scalar number r_u and calculates the EC point ($R_U = r_u G$) corresponding to the selected number. Then, the U sends the information that needs to be in the certificate, $Cert_{info}$, and R_U to the CA. Then, CA also selects a random number c and calculates the public parameter of the certificate, $P_U = cG$. After that, the CA generates the certificate $Cert_U$ using $Cert_{info}$ and P_U and calculates the hash of the $Cert_U$, e. Finally, CA calculates the private key contribution of the CA, a, and sends back a and $Cert_U$ to the U. U can calculate its private key, $d_U = er_u + a$, and public key $Q_U = eP_U + Q_{CA}$ as shown, where (d_{CA}, Q_{CA}) is the private and public key of the CA.

C. Blockchain

Blockchain is a decentralized and distributed ledger technology that acts as the underlying technology of several digital cryptocurrencies [17]. Decentralization, transparency,

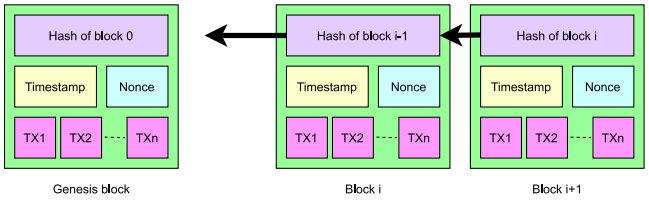

Fig. 7. A sample blockchain network.

immutability, and auditability are some of the key features of blockchain technology. Due to these different advanced features, blockchain is utilized in several applications, such as healthcare facilities, risk management, and social services [18]. Moreover, blockchain has gained significant attention in the research community to develop security applications due to its features.

Typically, a blockchain network consists of a sequence of cryptographically bound blocks. Figure 7 illustrates an example blockchain network. Each block points to the immediate previous block, called the parent block,by storing the hash of the corresponding block. Therefore, if anybody changes a previous block in the network, he has to alter every block after that block and all the nodes which maintain a copy of the ledger in the network. The first blockchain block, the genesis block, has no parent. A smart contract is a computer program stored in the blockchain that can be used to implement custom logic to implement some operations. In this paper, the Hyperledger Fabric blockchain network, which is an open-source blockchain platform, is used in the implementations.

D. Related Works

Authentication and authorization are a couple of key security aspects in telecommunication networks. As per the 3GPP, future telecommunication networks use certificates for mutual authentication, and they use RE presentational State Transfer (REST) for the internal communications between NFs [19]. However, the certificate management processes, such as certificate issuance, cross-domain trust, certificate validation, and certificate revocation, are not clearly described in the specifications. Traditional PKI architecture has several limitations when applied to telecommunication networks. Using a commercial CA to handle certificates would be a very costly operation as deploying NFs and network slices is a frequent activity in telecom networks. If we use an internal CA to handle certificates, communication between NFs that are deployed in multi-domain network slices is not possible, as certificates are only visible to the internal network.

The various iterations of ECC in multi-party computation schemes and multi-party signing protocols have been explored in existing literature. In [20], a highly efficient non-interactive key-exchange (NIKE) mechanism is introduced. This mechanism enables the generation of a shared key within a group by leveraging contributions from its members. Additionally, in [21], Payeras et al. present a multi-party contract signing framework that eliminates the need for a trusted third party by utilizing ECC and blockchain technologies. Furthermore, Dai et al. propose a robust three-factor authentication scheme tailored for wireless sensor networks, employing ECC [22]. This scheme is specifically designed for a multi-gateway environment, aiming to minimize the number of authentication messages while maintaining the desired security standards. Although there are existing implementations of multi-party signing protocols and key-sharing schemes utilizing ECC, none seem to incorporate a single key generation mechanism leveraging contributions from multiple parties. To the best of our knowledge, as of the time of conducting this research, the key generation mechanism proposed in this paper, utilizing ECC and an extended version of ECQV with multi-party contributions, represents the first such implementation documented in the literature.

A blockchain is an emerging approach to implementing PKI frameworks in existing research. In [23], Kubilay et al. proposed a blockchain-based PKI framework known as CertLedger to eliminate the split-world attack and to provide certificate/ revocation transparency. Trusted CA certificate management, validation, revocation, and storage are managed through the blockchain in their platform. In [24], Hewa et al. proposed a framework to manage ECQV certificates for IoT devices through a blockchain platform. Blockchain is used to simplify the certificate issuance and revocation processes in this research. ProofChain is a blockchain-based PKI framework that was proposed to select a CA randomly from a certificate pool to handle certificates [25]. In [26], Yakubov et al. proposed a PKI framework that can issue an extended version of X.509 certificates. Khieu and Moh introduced an architecture for Public Key PKI hosted in the cloud, leveraging blockchain technology to establish secure access to certificate data and revocation lists [27]. In [28], Adja et al. presented a system for managing certificate revocation and verifying certificate statuses using blockchain technology, with a specific focus on X.509 certificates and their extension while utilizing the blockchain ledger for revocation information storage. Luo et al. proposed a blockchain-based PKI framework with a focus on scalability, employing redactable blockchains to address certificate revocation challenges [29]. Building upon Luo's architecture, ChainPKI, as discussed in [30], was designed to address privacy concerns within PKI frameworks. However, all these existing works rely on a single-parent CA, and if it is vulnerable, it impacts all the users in the network. Also, this CA is a single point of failure.

A framework for managing certificates using a blockchain-based PKI is proposed in [31]. Also, they have proposed an optimization method to improve certificate storage efficiency and revocation simplicity. In [32], Yan et al. extended the framework in [31] to perform a decentralized certificate management framework in an NFV environment. However, these systems are also subject to the mentioned drawbacks in existing PKI frameworks.

3GPP specifies that the authorization between NFs is handled by the NRF using Oauth2 access tokens [8]. However, as highlighted in [10], this system is vulnerable to the above-mentioned security attacks when the NFs are shared between network slices. At the time this research is being conducted, to the best of our knowledge, no existing research can be found to mitigate these vulnerabilities in the authorization process.

TABLE I
FEATURE COMPARISON WITH KEY RELATED WORKS

Features	[18]	[20]	[19]	[21]	[26]	[27]	Ours
Distribution of CA functionality to increase trust	✗	✗	✗	✗	✗	✗	✓
Mitigation of single point of failure due to centralized CA	✗	✓	✗	✗	✗	✗	✓
Light weight certificates	✗	✗	✓	✗	✗	✗	✓
Fast and automatic certificate revocation	✓	✓	✓	✓	✓	✓	✓
Telecom specific implementation	✗	✗	✓	✗	✓	✓	✓
Secure communication protocols	✗	✗	✗	✗	✗	✗	✓
Mitigation of unauthorized service utilization	✗	✗	✗	✗	✗	✗	✓
Elimination of DoS attacks due to malicious NFs	✗	✗	✗	✗	✗	✗	✓
Mitigation of data leakage between slices	✗	✗	✗	✗	✗	✗	✓

E. Comparison Table With Existing Works

Table I shows the feature-wise comparison of the proposed certificate issuance framework with existing proposed frameworks. The key-related works were selected considering the relevance to our approach, implementation details, and experimental results. The feature list is extracted from the proven results of the experiment. From the comparison of the Table I, we can deliberate that our certificate issuance framework outperforms, considering other related works.

III. PROPOSED FRAMEWORK

In this section, we present our proposed frameworks, along with the considered threat model and security features, for the certificate issuance process in telecommunication networks and for authorization between NFs for managing access tokens. While the certificate issuance framework targets certificate management in telecom networks, we mainly considered the mitigation of mentioned security attacks in the authorization framework. Table II denotes the used notations throughout the paper.

A. Threat Model

During the certificate creation, we take the assumptions of the Dolev and Yao [33] threat model in order to depict the robustness of the designed decentralized certificate management framework for mutual authentication. This model suggests that an attacker (α) is capable of carrying out a variety of tasks.

- During the production of certificates at the MNO and NF levels, α has the potential to intercept the messages being exchanged.
- α has the potential to replay the intercepted exchanged message.

TABLE II
NOTATIONS

Symbol	Definition
O_R	Certificate requested operator
C_O	Operator specific smart contract
C_N	Network function specific smart contract
F_p	Finite space for the EC
a, b	Coefficients of EC
G	Generator of EC
H	Hash function
$HMAC$	Hash Message Authentication Code
$Cert_R$	Certificate for requested party
$Cert_{info}$	Information for the certificate
e	Certificate hash
P	Public certificate parameter
r	Random scalar for EC operations
r_R	Random scalar selected by the requester
R	EC point generated using r, i.e. $R = rG$
R_R	EC point generated using r_R
$\{d_B, Q_B\}$	Key pair for the communication with blockchain
$\{d_{O:k}, Q_{O:k}\}$	Key pair of operators $k \in 1, 2, .., n$
T	Timestamp
X	Random number

- α has the potential to acquire the long-term secrets used in the production of certificates at the MNO and NF levels.
- α has the potential to interlink the exchanged messages of various successful certificate creations in order to trace the location of MNOR.
- α has the potential to impersonate the entity involved in the production of certificates at the MNO and NF levels.

B. Security Features

The following security features [34], [35] are verified during the production of certificates at the MNO and NF levels.

- Resilient against replay attack: Previous session messages that have been captured prevent the attacker from sending the same message again to obtain a response.
- Resilient against traceability attack: In order to track down the location of the entity engaged in the generation of certificates at the MNO and NF levels, the attacker will not be able to interlink messages from prior successful sessions that have been captured.
- Resilient against impersonation attack: It is impossible for the attacker to impersonate the entity involved during the production of certificates at the MNO and NF levels.
- Resilient against non-repudiation: During the production of certificates at the MNO and NF levels, the entity involved can not deny that they have not applied for the certificate.
- Resilient against DoS attack performed through intercepted messages: Captured messages of previous sessions will not allow the attacker to resend the message at very high speeds or volumes in order to jam the network or to disturb the network.
- Perfect forward secrecy: Acquiring long-term secrets will not allow the attacker to derive the private key during the certificate creation.
- Stolen verifier attack: Acquiring the secrets stored on the blockchain and the security orchestrator will not allow the attacker to derive secret information of the certificate creation.

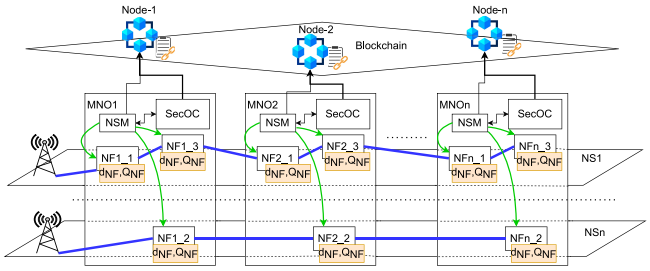

Fig. 8. Considered environment for the proposed certificate management framework.

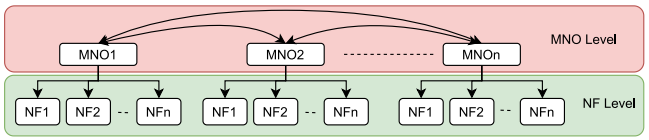

Fig. 9. Certificate hierarchy followed by the proposed framework.

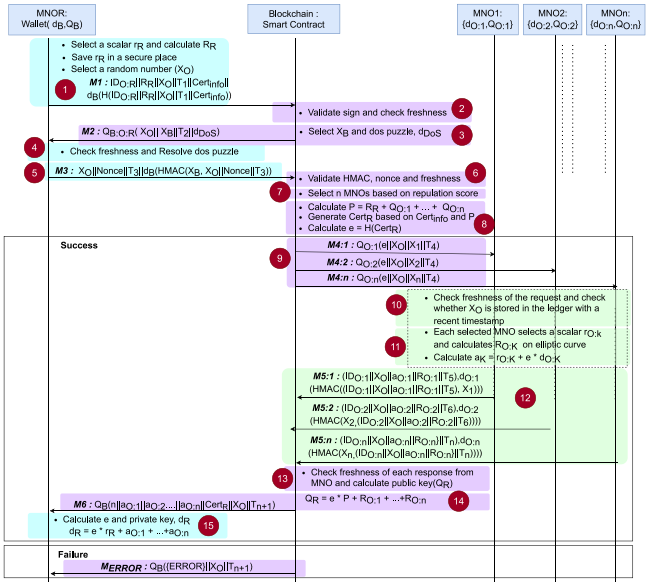

Fig. 10. Security flow diagram for the MNO certificate generation.

C. Decentralized Certificate Management Framework for Mutual Authentication

According to the 3GPP specifications, Transport Layer Security (TLS) certificates are employed to mutually authenticate different network functions. In this paper, we designed a blockchain-based decentralized certificate management framework to issue, validate, and revoke certificates in an environment that consists of network slices created by multiple network operators. Figure 8 shows the considered environment with relevant entities. In our solution, we eliminate the requirement of a central trusted entity as the root of the certificate chain for the certificate management process. A decentralized trust-building mechanism is proposed to perform the functionality of the root certificate authority.

As we build trust using multiple parties in the network, communication with multiple parties is required when issuing a certificate. This is a costly operation to follow all the time when issuing certificates. Therefore, only MNO-wise certificates are issued using the distributed trust mechanism, and as NFs are internal entities in an MNO environment, the MNO can manage certificates of NFs. Thus in our solution, we maintain certificates in two levels, i.e., MNO level and NF level (Figure 9). Implicit certificates with elliptic curve cryptography are utilized to provide authenticity, confidentiality, integrity and non-repudiation in the communication in the considered environment.

1) System Initialization: The functionality of our framework depends on a blockchain network. A private, public, or consortium blockchain can be utilized to create our framework. The users are required to be able to communicate with the blockchain network. We assume that all the entities in the blockchain network possess a public-private key pair with them, as almost all the available blockchain networks employ such a mechanism for communication between the blockchain and the user. For simplicity, we assumed that there are only telecommunication-related entities such as network operators, service providers, Local 5G Operators (L5GOs), and infrastructure providers are available in the network. However, we can use any type of user who is willing to participate in the certificate generation process in the blockchain with our framework.

2) MNO Certificate Generation: The considered environment consists of multiple MNOs, and we consider them at the highest level of the certificate trust hierarchy. Initially, the domain parameters of the elliptic curve $\{F_P, a, b, G\}$ are distributed in the blockchain ledger. When a particular MNO requires a certificate, it sends a certificate-issuing request to the smart contract in the blockchain. Then, the smart contract executes the group trust-building mechanism to generate the certificate. The procedure for MNO certificate generation is shown in Figure 10.

- *Step 1:* In order to request a certificate by MNO, O_R, selects an initial random scalar r_R and computes the EC point, $R_R = r_R G$, associated with the established EC domain parameters in the blockchain. Then O_R selects a random number X_O and certificate information $Cert_{info}$. $Cert_{info}$ contains the details that need to be in the certificate, such as MNO id, MNO name, Mobile Network Code (MNC), and Mobile Country Code (MCC). After that, O_R originates the certificate request which contains the identity of the O_R, $ID_{O:R}$, X_O, R_R, and the current timestamp T_1. O_R calculates the hash of the complete message and signs it using the wallet private key, d_B. The first message from O_R to the C_O is as follows.

$$M1 = ID_{O:R}||R_R||X_O||T_1||Cert_{info}|| \; d_B\big(H(ID_{O:R}||R_R||X_O||T_1||Cert_{info})\big) \quad (1)$$

  O_R keeps the r_R within itself in a secure place to compute the private key (d_R) in future steps.
- *Step 2:* After receiving the certificate request from O_R, C_O validates the signature of the message appended at the previous step using the public key, Q_B of O_R which is correspondent to the d_B in O_R's wallet. Also, C_O checks the freshness of the request using Equation (2).

T_2 is the current timestamp, and T is an agreed parameter in the system to check the freshness of a request.

$$T > T_2 - T_1 \tag{2}$$

- *Step 3:* C_O selects a random number X_B and a DoS puzzle parameter, d_{dos}, and sends M_2 back to the O_R. M_2 can be calculated as follows. This DoS puzzle parameter supports mitigating the potential DoS attacks on other network operators in the blockchain network as the C_O sends requests to a selected set of operators for each certificate request.

$$M2 = Q_B(X_O||X_B||T_2||d_{dos}) \tag{3}$$

- *Step 4:* Upon decryption of the received request using d_B, O_R checks the freshness of the message using Equation (4). T_3 is the current timestamp. Then O_R solves the received DoS puzzle. O_R calculates a nonce so that the preceding d_{dos} number of bits of the hash of the $X_O||X_B||nonce$ is equal to zeros.

$$T > T_3 - T_2 \tag{4}$$

- *Step 5:* O_R generates $M3$ using Equation (5). The message contains X_O, *nonce*, T_3, and the signed hash MAC of the message. X_B is used as the key in the hash MAC calculation.

$$M3 = X_O||nonce||T3||d_B(HMAC(X_B, X_O||nonce||T3)) \tag{5}$$

- *Step 6* C_O checks the freshness of the received message with current timestamp $T4$ using Equation (6), validates the hash MAC and confirms whether the received nonce resolves the sent dos puzzle.

$$T > T_4 - T_3 \tag{6}$$

- *Step 7:* C_O selects n number of MNOs in the blockchain network based on the reputation score. The reputation score for a particular MNO, O_{rep}, can be calculated using Equation (7). It is based on the number of NF certificates of the MNO (N_{owned}), the number of contributions for certificate issuance $(N_{contributed})$, and the number of revoked certificates in the MNO network $(N_{revoked})$. After arranging the reputation scores in descending order, a number of n MNOs with the highest reputation scores are selected for the certificate generation procedure. The n depends on the number of available MNOs in the blockchain network.

$$O_{rep} = N_{owned} + N_{contributed} - N_{revoked} \tag{7}$$

- *Step 8:* $Cert_R$ is generated using the received $Cert_{info}$ and the public parameter, P for a particular certificate is calculated using the public keys of the selected MNOs and received parameter from O_R (Equation (8)). Then the hash of the $Cert_R$ and P, e is calculated (Equation (9)).

$$P = R_R + \sum_{i=1}^{n} Q_{O:K} \tag{8}$$

$$e = H(Cert_R, P) \tag{9}$$

- *Step 9:* A random number X_k is selected and $M4{:}k$ is calculated and sent to each selected MNO. Each message is encrypted using the public key, Q_k, of the selected MNOs. T_4 is the current timestamp. $(k \in 1, 2, \ldots, n)$

$$M4{:}k = Q_{O:k}(e||X_O||X_k||T_4), \quad k \in 1, 2, \ldots, n \tag{10}$$

- *Step 10:* Each MNO decrypts the received message using its private key, validates the freshness of the received message, and confirms that the X_O is stored in the blockchain ledger. This ensures that the request is sent by the blockchain network.
- *Step 11:* Then each MNO selects a random scalar $r_{O:k}$ and calculates the $R_{O:k}$ considering the established EC domain parameters. Then, each MNO calculates the certificate contribution parameter $a_{O:K}$ as shown in Equation (11). $d_{O:k}$ is the private key of each MNO.

$$a_{O:k} = r_{O:k} + e * d_{O:k}, \quad k \in 1, 2, \ldots, n \tag{11}$$

- *Step 12:* $M5{:}k$ is calculated by each MNO, considering $ID_{O:k}$, timestamp t_k, X_O, and $\{a_{O:k}, R_{O:k}\}$ (Equation (12)). Each hash mac of $M5{:}k$, which is calculated using the received X_k as the key, is signed using the $d_{O:k}$. T_k is the current timestamp.

$$\begin{aligned} M5{:}k = {} & (ID_{O:k}||X_O||a_{O:k}||R_{O:k}||T_k)|| \\ & d_{O:k}(HMAC(X_k, (ID_{O:k}||X_O||a_{O:k}||R_{O:k}||T_k))), \\ & k \in 1, 2, \ldots, n \end{aligned} \tag{12}$$

- *Step 13:* Upon receiving the $M5{:}k$ from each MNO, C_O validates the freshness of the responses. Also, C_O confirms that the hash MAC is calculated using the sent X_k and signed using the $d_{O:k}$.
- *Step 14:* C_O calculates the public key, Q_R of the O_R as follows, generates $M6$, and shares it with O_R

$$Q_R = e * P + \sum_{i=1}^{n} R_{O:K} \tag{13}$$

$$M6 = Q_B(n||a_{O:1}||\cdots||a_{O:n}||Cert_R||X_O||T_{n+1}) \tag{14}$$

- *Step 15:* Finally, O_R validates the freshness of the received message and calculates e and the private key, d_R using the parameters in the received message. X_O is used to identify that the messages belong to the same certificate generation process.

$$d_R = e * r_R + \sum_{i=1}^{n} a_{O:K} \tag{15}$$

If any of the intermediate steps fail, O_R receives M_{Error} (Equation (16)).

$$M_{Error} = Q(\{ERROR\}||X_O, T_{n_1}) \tag{16}$$

The proof of the distributed multi-party private key and public key generation procedure is as follows.

$$Q_R = e_R * P_R + \sum_{i=1}^{n} R_i \tag{17}$$

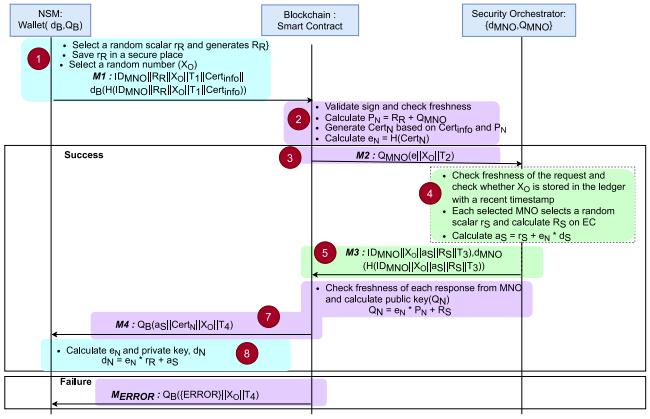

Fig. 11. Security flow diagram for the NF certificate generation.

$$= e_R * \sum_{i=1}^{n} Q_i + \sum_{i=1}^{n} R_i + e_R * R_R \tag{18}$$

$$= e_R * \left(\sum_{i=1}^{n} d_i * G + r_R * G \right) + \sum_{i=1}^{n} r_i * G \tag{19}$$

$$= \left(\sum_{i=1}^{n} d_i * e_r + r_i + e * r_R \right) * G \tag{20}$$

$$= \left(\sum_{i=1}^{n} a_i + e * r_R \right) * G \tag{21}$$

$$= d_r * G \tag{22}$$

3) NF Certificate Generation: When a new NF is deployed, a certificate needs to be generated and stored inside it for the NF's communication purposes. The Network Slice Manager (NSM) generates the required certificates with the support of the blockchain and the security orchestrator (SecOC) in the MNO. Figure 11 shows the procedure for generating certificates for NFs.

- *Step 1:* The NSM selects an initial scalar r_R and calculates a point,$R_R = r_R G$ on the EC. Then, $Cert_{info}$ is generated, including all the required information of the new NF. The initial message M_1 is produced concatenating the identity of the MNO (ID_{MNO}), R_R, a random number (X_O), and the current timestamp (T_1). The M_1 is signed using the wallet private key (d_B).

$$M1 = ID_{MNO}||R_R||X_O||T_1||Cert_{info}||$$
$$d_B\big(H(ID_{MNO}||R_R||X_O||T_1||Cert_{info})\big) \tag{23}$$

  The NSM keeps the r_R within itself securely to calculate the private key of the new NF, d_N.
- *Step 2:* Smart contract-related issuing certificates for NFs, C_N receives the M_1, and it confirms that the signature is valid and the message is fresh (Equation (24)). Then C_N generates the NF certificate, $Cert_N$, including the details in $Cert_{info}$. The public parameter of the $Cert_N$, P_N, and the hash of the $Cert_N$, e_N, are calculated using and 26, respectively.

$$T > T_2 - T_1 \tag{24}$$
$$P_N = R_R + Q_{MNO} \tag{25}$$
$$e_N = H(Cert_N, P_N) \tag{26}$$

- *Step 3:* C_N originates the M_2 including the e, X_O, and current timestamp T_2 and sends the M_2 to the security orchestrator, *SecOC*, of the MNO. The content of the message is encrypted using the public key of the MNO, Q_{MNO}.

$$M2 = Q_{MNO}(e||X_O||T_2) \tag{27}$$

- *Step 4:* The *SecOC* decrypts $M2$ using the private key of MNO, d_{MNO}, and validates the freshness of the message (Equation (28)). T_3 is the current timestamp. Then it selects an EC point, $\{r_S, R_S\}$, considering the curve parameters in the blockchain. The private key contribution parameter, a_S, is calculated as shown in Equation (29).

$$T > T_3 - T_2 \tag{28}$$
$$a_S = r_S + e * d_{MNO} \tag{29}$$

- *Step 5:* After calculating the *MNO* contribution for the NF key pair generation, the *SecOC* originates $M3$. $M3$ contains $\{a_S, R_S\}$, ID_{MNO}, X_O, and T_3. The content of the message is signed using the d_{MNO}.

$$M3 = ID_{MNO}||X_O||a_S||R_S||T_3||$$
$$d_{MNO}(H(ID_{MNO}||X_O||a_S||R_S||T_3)) \tag{30}$$

- *Step 6:* Upon receiving the $M3$, C_N validates the freshness of the message and verifies the signature. Then, it calculates the public key of the network function, Q_N, using equation (32) and stores it in the ledger.

$$T > T_4 - T_3 \tag{31}$$
$$Q_N = e_N * P_N + R_S \tag{32}$$

- *Step 7:* C_N generates the $M4$, which contains the parameters required to calculate the private key, d_{NF}, of the new NF. Additionally, $M4$ consists of X_O, and the current timestamp T_4, and it is encrypted using the Q_{MNO}.

$$M4 = Q_{MNO}(a_S||Cert_N||X_O||T_4) \tag{33}$$

- *Step 8:* Finally, the NSM decrypts $M4$ and validates the freshness of the message as shown in Equation (34) (T_5 is the current timestamp). Then, it calculates the hash of the $Cert_N$, e_N, and the private key of the new NF, d_N as follows.

$$T > T_5 - T_4 \tag{34}$$
$$d_N = e_N * r_R + a_S \tag{35}$$

If any error occurs in the procedure, the NSM receives M_{ERROR} (Equation (36)). X_O is used to identify the messages in a single NF certificate generation procedure.

$$M_{ERROR} = Q_B(\{ERROR\}||X_O||T4) \tag{36}$$

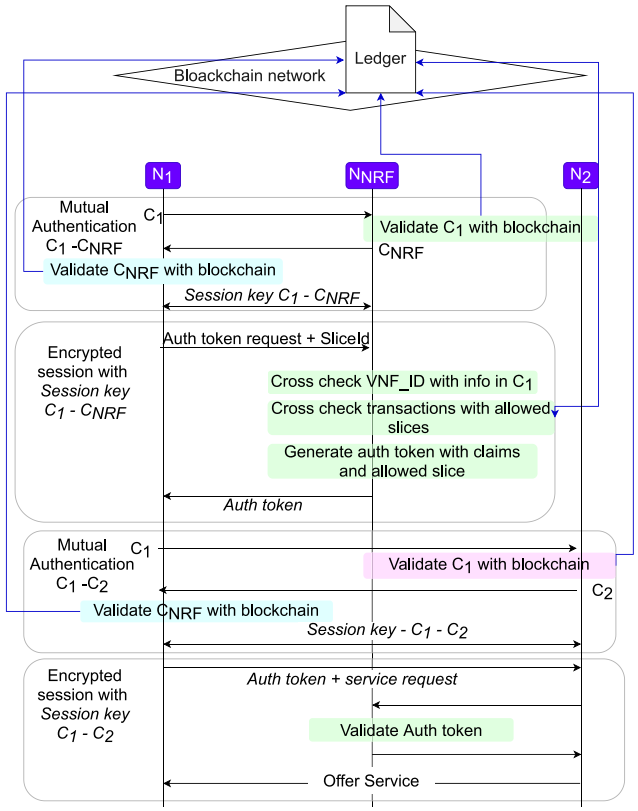

Fig. 12. The proposed framework for acquiring the authorization for inter-NF communication.

D. Blockchain-Based Authorization Framework

The authorization between NFs is handled by the Network Repository Function (NRF) in traditional 5G networks. There are two methods for communication between NFs in 5G networks: direct communication, which is communication directly while receiving an access token from NRF, and indirect communication, where communication is through a Service Communication Proxy (SCP), which gets the access token from the NRF [9]. However, this system is vulnerable to mentioned security attacks when NFs are shared between network slices. Therefore, we designed a novel authorization framework for 5G and beyond networks with the support of blockchain.

When a particular NF is shared between network slices or deployed in a network slice, NSM generates a transaction request which includes the NF id and the corresponding slice id and sends it to the blockchain network. We can use the same blockchain network that we used to generate certificates earlier here. This information can then be used when providing authorization between NFs, which are shared between multiple network slices across multiple administrative domains.

Figure 12 shows the proposed authorization framework for the direct communication between NFs. Here, N_1 denotes the consumer NF and N_2 denotes the producer NF. Initially, the N_1 mutually authenticates with the NRF using the certificates generated through our certificate framework. After verifying the certificates through the blockchain, N_1 and N_{NRF} share the session key securely using the private-public key pairs. Then, acquiring the access token process is encrypted using the shared session key. N_1 sends the access token request, which includes a slice id. Upon receiving the request to *NRF*, it performs regular authorization checks to check whether the N_1 can access services in N_2. In addition, it queries the blockchain to check whether the N_1 is actually deployed in the received slice. If *NRF* can find such a transaction and if other authorization checks are passed, *NRF* generates an access token, which is a JSON Web Token (JWT) including N_1 id, N_2 id, and slice id as claims in the token in addition to other required claims.

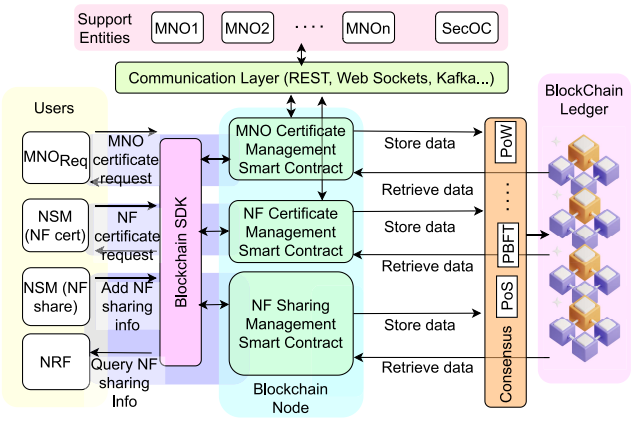

Fig. 13. Blockchain deployment of the proposed framework.

After receiving the access token from *NRF*, N_1 mutually authenticates with the N_2 following the same procedure for the authentication between N_1 and N_{NRF}. Then, N_1 sends the service request to the N_2 with the received access token. N_2 validates the access token with the support of *NRF*, and if the validation is successful, it offers the service to N_1.

This procedure covers the authorization for direct communication. However, with slight modifications, we can use the same procedure for indirect communication.

E. Deployment of the Framework in Blockchain

This framework can be deployed in a blockchain network to ensure trust and immutability. Figure 13 illustrates deploying the proposed framework in a blockchain network. A public, private, or consortium blockchain that supports smart contracts can be used with our framework, as we are not storing any sensitive information, such as private keys, in the blockchain. However, we have to follow the blockchain network-specific configurations when selecting an appropriate network, such as policies in the network. The consensus mechanisms, such as Proof of Work (PoW), Proof of Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT), can be used with our framework according to the selected blockchain network. Mining is an essential functionality in blockchain networks. If we use a public blockchain, the existing miners in the network can also be used as miners in our framework. If we use a private blockchain deployed among MNOs, MNOs can be the miners in this network, as the functionality benefits all the MNOs in the network.

Primitively, we need to implement three smart contracts to implement the functionality of our framework. The functions related to these smart contracts are described earlier.

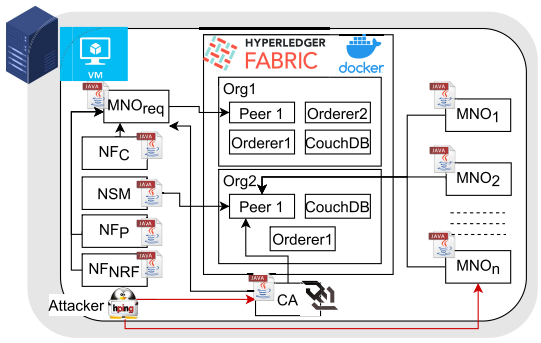

Fig. 14. Testbed implementation with Hyperledger Fabric.

Users communicate with the smart contracts using blockchain-specific Software Development Kits (SDKs), and smart contracts may need to communicate with external parties already members of the blockchain. The smart contract responsible for MNO certificate management must select and communicate with n MNOs to collect the contributions for MNO certificate generation. NF certificate management contract communicates with SecOC to generate a certificate for an NF. NF sharing management contract does not require communication with external parties for its functionality. Also, we must add all the requests and responses from external parties into the blockchain to ensure accountability and transparency. Therefore, the blocks that are created in the blockchain with our framework should contain transactions related to inbound and outbound requests and responses, issued certificates and their status (revoked or not), and NF sharing details among slices.

IV. Experimental Analysis

In this section, we provide details about the implementation of our testbed. Also, an analysis of the performed set of comprehensive experiments can be found at the end of this section.

A. Implementation Setup

We used Hyperledger Fabric (Version 2.1.0) to build a Proof of Concept (PoC) of our proposed decentralized certificate management framework. We have considered a blockchain network that consists of two organizations in the experiments. One organization has one peer node and two ordered nodes, and the other organization has one peer node and one orderer node. Peer nodes and orderer nodes are running as docker containers, and CouchDB is used as the state DB. The applications related to certificate requests, smart contracts, and responses for certificate requests are developed using Java and Web sockets. For comparison, a CA is developed using Java, and it is used with the same Hyperledger network, considering proposed architectures in existing certificate management frameworks. The size of the DoS puzzle used in the experiments is five zeros at the front of the generated hash. We implemented this framework in a machine with 16GB RAM and 8 CPU cores.

In our implementation, we developed the three smart contracts as we described in section III-E. One is for handling

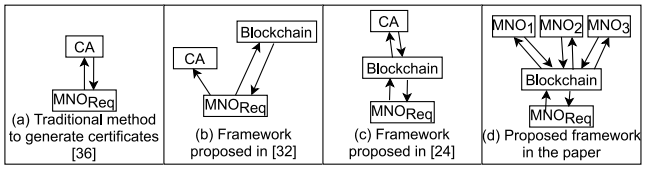

Fig. 15. Benchmarked architectures in the experiment.

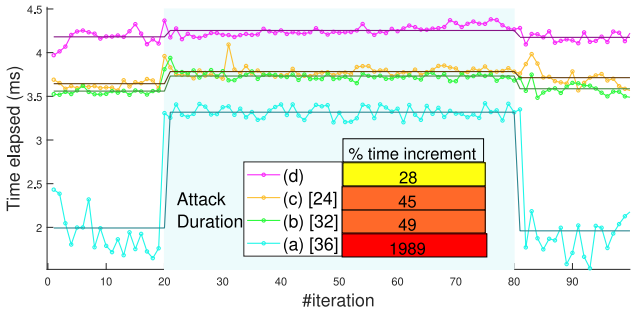

Fig. 16. The impact of a DoS attack on the certificate issuer.

the certificates for the MNOs, one is for handling the process of NF certificates, and the other is for handling NF sharing between network slices. Users (MNOs, NSM, and NRF) communicate with smart contracts deployed in peers. Orderers handle the process of ordering the transactions and distributing them across the network. All the experiments were performed several times, and averages were taken to eliminate inconsistencies in the results.

B. Performance Analysis Experiments of the Proposed Certificate Management Framework

Here, we experimented with the functionality and the performance of the proposed certificate framework. The impact of the multi-party contribution and performance analysis when processing batches of certificates are analyzed here.

1) Performing a DoS Attack on the Certificate Issuing Party: In this experiment, we investigated the impact of a DoS attack on the certificate issuance process. We have considered four scenarios in this experiment, i.e., a traditional CA-based system without blockchain [36], a single CA entity with blockchain as proposed in [32], a blockchain between the CA and certificate requested party [24], and our framework (Figure 15). We performed a DoS attack on the certificate issuer during the 20th and 80th iterations and measured the time to receive a certificate from the system. As we performed the DoS attack on a single entity in all the systems in our framework, we randomly selected an MNO and performed the attack on it. Four MNOs were deployed in the network, and one MNO was subjected to the DoS attack. We considered three MNOs randomly for certificate generation.

Figure 16 shows the received results in this experiment. Architecture (a) shows the minimum time to issue a certificate, as blockchain interaction is not required to issue certificates. Architecture (b) and architecture (c) show nearly equal values to issuing certificates due to the interaction with the blockchain. Our framework takes the highest time to issue certificates due to the involvement of multiple parties to issue a certificate. During the DoS attack, architectures (a), (b),

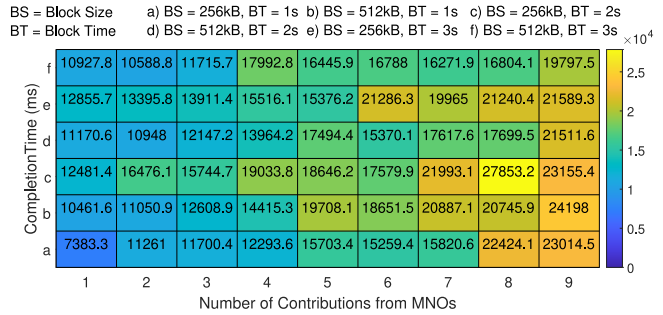

Fig. 17. Consumed time for receiving an MNO certificate.

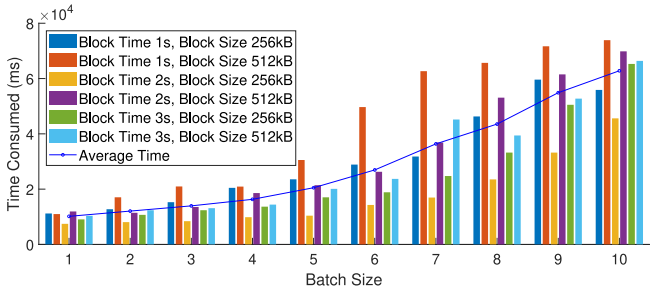

Fig. 18. Consumed time for a batch of NF certificates.

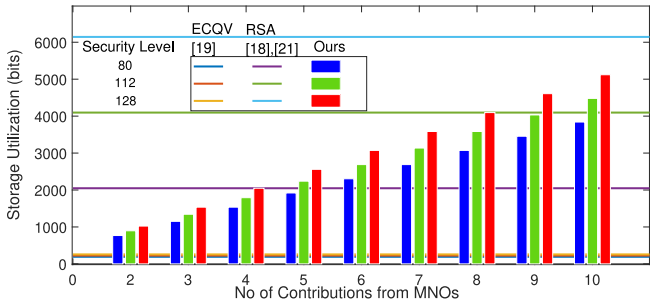

Fig. 19. Storage utilization for storing certificates.

and (c) are subject to a nearly equal impact as they all depend on a single CA entity. However, proportionally (table in Figure 16), architecture (a) has the highest performance degradation when compared with normal time as in normal time, it consumes a very short time to generate a certificate. Even though architecture (b) and (c) faced approximately equal impact, when comparing the proportion, it is a lower value. Our architecture performed well during the DoS attack, as it does not depend on a single CA or entity to issue a certificate. The time consumed during the attack in our architecture can be further reduced by increasing the participating parties when there is a higher number of parties in the network.

2) Time to Issue an MNO Certificate: Here, we analyzed the time taken to issue an MNO certificate (root certificate for an operator) using our framework. We have increased the number of participating other operators and measured the time elapsed to generate a certificate. We also performed the experiment with the different settings of the blockchain network to assess the impact of these configurations on performance.

As shown in Figure 17, when the number of participating MNOs increases, the time taken to issue a certificate increases. An increment in the communication time needed to collect certificate contributions is the cause of this observation. The alterations of parameters in the blockchain network are not affected significantly by the results as shown in the graph Fig. 10. Fluctuations can be observed due to the time elapsed to resolve the dos puzzle, as the receiving dos puzzle is completely random in each request.

3) Time to Issue an NF Certificate: Issuing certificates for NFs is a frequent operation in an operator network. In this experiment, we investigated the time complexity of issuing our system certificates for batches of NFs. We submitted certificate requests as batches to our system, and we increased the batch size and measured the time elapsed to generate certificates for a particular batch during the experiment. The configuration parameters, such as block time and size, are altered, and experiments are performed with different settings in the blockchain network.

The time consumed with each blockchain setting and the average time for all the different blockchain settings is shown in Figure 18. As we increase the number of certificate requests in a batch, the time consumed to complete the total process increases with all the different blockchain settings. However, each higher block size (512 *kB*) corresponding to the same block time shows a higher completion time. A block is not created until it receives the configured size of transactions or until the block time elapses. When the block size is large, the fabric should receive a higher number of transactions or transactions with increased size. Even if it receives some transactions, a new block is not created until the block time expires. However, when the block size is small, it creates blocks rapidly, and the created blocks are available in the blockchain before the block time elapses. This causes the observation of higher time consumption for larger block sizes. The exponentially increasing nature of the graphs is observed due to the increased amount of parallel processing required in the certificate-issuing application.

4) Ledger Storage Utilization for Storing Generated Certificates: Here, we have investigated the ledger storage utilization for storing the generated certificates with our framework, an extended version of the ECQV certificates against state of the art. Traditional ECQV certificates and RSA certificates are considered state-of-the-art in this evaluation. The certificate size in our framework is a bit larger than traditional ECQV certificates due to the number of contributions, and they are considerably smaller than RSA certificates. Blockchain-based implementations are evaluated in this analysis. For instance RSA - [23], [26], ECQV - [24]. The simulation parameters are extracted from the [37]. Also, we considered different security levels for these different certificate types in this simulation for more elaborate results.

Figure 19 illustrates the received results of this investigation. As there is no impact from the number of contributors for ECQV and RSA certificate generation, the storage utilization in the ledger is constant with these certificate types. Initially, when the number of contributors is low, the storage utilization for a particular certificate in our framework is considerably lower than RSA certificates and slightly higher than traditional

TABLE III
SYSTEM BEHAVIOR FOR SERVICE UTILIZATION ATTACK

	Authorization Time	Attack mitigation
Our system	$1798 \pm 352.86ms$	Yes
Traditional method	$100.55 \pm 40.74ms$	No

ECQV certificates. However, when the number of contributions rises, storage utilization continuously increases, and when it is very high, storage utilization exceeds the storage required for RSA certificates. This observation can be seen due to the requirement to store the contributions from different MNOs for certificate generation with our framework. Also, we can see that when the security level increases, storage utilization increases in all three certificate types.

C. Experiments on NS-Related Attacks Mitigation

Here, we investigate how our framework mitigates the discussed three NS-related attacks in the background section (Section II). We show that the proposed improvements for the NF authorization in our framework can eliminate all three attacks.

1) Mitigation of Unauthorized Service Utilization: Here, we experimented with how our framework can mitigate unauthorized service utilization attacks. The direct communication scheme is used in this experiment, and authentication is performed using certificates. As shown in Figure 2, the cause for this attack is the NRF cannot check whether the malicious NF sends the correct slice id or not. However, with our system, when a particular NF is shared with a slice, that record is added to the blockchain. Then, before providing the access token, the NRF cross-checks the received slice id and the stored slice id in the blockchain against the NF id. If it matches, it provides authorization; otherwise, the request is denied. Therefore, our system can successfully mitigate this attack. In this experiment, we first configured the blockchain to store the VNF deploying details. Then, we sent service requests to the shared NF while misconfiguring the slice ids in the authorization request following the direct communication scheme.

Table III shows the received results in this experiment. The experiment is performed around 50 times, and the time for authorization is calculated with 95% confidence. The time consumed for authorization in our system has increased considerably compared with the traditional authorization system. We received an authorization token for legitimate requests, and requests are denied when we send malicious requests with an unauthorized slice id. Even though there is an apparent increase in the authorization time in our system, it could mitigate all the unauthorized service utilization attacks in the slicing ecosystem. Therefore, slice owners of the victim slice do not want to pay for unused services.

2) Elimination of DoS Attacks Due to Malicious Overload-Control Header: In this attack, even though malicious NF takes authentication, it can send the overload control header related to another slice for a shared NF, as shown in Figure 4. As we include the authorized slice into the authorization token in our framework, the shared NF can cross-check and validate the header. In this experiment, the malicious NF in slice two sends the overload control header related to slice one in the 30th iteration, and then it notifies the shared NF that slice one is back to normal in the 70th iteration. In this experiment, we measured the processed traffic percentages related to each slice through the shared NF. Here, we assume in normal conditions, slice one and slice two follow poison distribution when sending requests to the shared NF, and they send requests with $\lambda = 30$ and $\lambda = 70$ (λ is the arrival rate in the poison distribution) in respective slices.

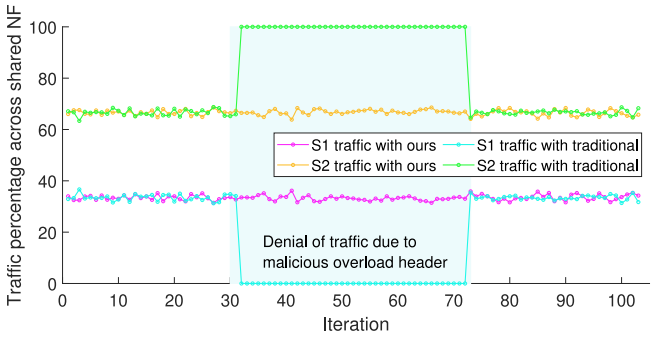

Fig. 20. The impact of traffic flow through the shared NF due to the overload control header.

TABLE IV
SYSTEM BEHAVIOR FOR THE DATA LEAKAGE BETWEEN SLICES ATTACK

	Authorization Time	Attack mitigation
Our system	$1234 \pm 191.51ms$	Yes
Traditional method	$102 \pm 50.13ms$	No

As shown in Figure 20, in the reference system, during the 30th and 70th iterations, shared NF stops the interactions with slice one and only interacts with slice two. In our system, we generate a JWT token that contains the authorized slice id as claims. Therefore, the shared NF can cross-check the slice id in the overload control header with the slice id in the authorization token. Thus, the shared NF disregards the overload control header and continues the services as usual in our system. Therefore, slice one would not be subjected to a DoS attack with our system due to the malicious NF in some other slice, as in the reference system. Due to the blockchain access to validate the received slice ID in the authorization request, this experiment also takes a longer time than the normal authorization with the NRF.

3) Mitigation of Data Leakage Between Slices: In this experiment, we tried to access the data of a user in some other slice. As shown in Figure 4, this attack also happens due to the inability of the NRF to validate the correct slice before issuing the authorization. However, our system makes all slice-sharing data against VNF ids available in the blockchain. The NRF validates the slice ids against VNF ids with the support of blockchain according to our framework. Even in a multi-domain network slicing environment, our system can mitigate these unauthorized slice data accesses.

Table IV shows the received results of this experiment. Our framework could eliminate all the unauthorized data access in network slices. We received authorization tokens only when we sent a valid slice id in the authorization requests into the NRF. However, the time consumed for establishing a successful session between two network functions is significantly greater than the traditional method due to blockchain queries.

V. Formal Analysis

The formal analysis section shows the security examination of the MNO certificate issuance and NF certificate issuance phases using the Oracle model known as Real-or-Random (ROR) logic [38], and widely adopted validation tool Scyther [39] used by many researchers to validate the security properties of the authentication protocol.

A. Formal Security Analysis Using ROR Logic

ROR is used in order to show that the attacker (α) will not get any secrets during the certificate generation at the MNO level and NF level. This logic was introduced by Abdalla et al. [38] to verify the protected communication protocols using cryptographic operations. Since both levels (MNO and NF level) use the same cryptographic parameters, we prove the security of the MNO level, which will also work for the NF level. During the certificate creation, the requesting MNO is called MNOR, Blockchain, and the *n* contributing MNOs, MNO1$\cdots$ MNOn, participate. These participants are represented by M^i, B^j, and O_n^k of the instances *i*, *j* and *k*. We use the same assumptions as taken by [38] and others papers [40], [41] during the proof that α can intercept exchanged messages and can perform the following acts such as editing and replaying in order to get the secrets used during the authentication. There are some queries defined thatα uses to get the secrets. The explanation of these queries is as follows.

- *Execute* $(M^i, B^j$ *and* $O_n^k)$*:* This query is executed by α to intercept the exchange messages between M^i, B^j and O_n^k.
- *Reveal* (E^h)*:* α executes this query to get the private key betweenM^i, B^j and O_n^k.
- *Send* (E^h, m)*:* α uses the above query to intercept the messages betweenM^i, B^j and O_n^k and this query is used to send the forged message to theM^i, B^j and O_n^k to get the response in order to determine the secrets.
- *Test* (E^h)*:* α uses this query to predict the actual secret key by tossing a coin *C*. It guesses on the basis of the outcome of the coin (i.e., $C = 0$, the communicating party returns the random number or $C = 1$, then the communicating party returns the private key. Otherwise, a null value is returned.)
- *Corrupt* (E^h)*:* α uses this query to obtain the secrets stored on the communicating entities involved in the certificate creation.

B. Semantic Security of the Session Key

This subsection shows that α can not get any secret information or random numbers used in the derivation of the private key of MNORduring the certificate generation from the captured exchange message between M^i, B^j and O_n^k.

Theorem 1: If α violates the semantic security of keys derived during the certificate generation in the proposed scheme, then $Adv_\alpha \leq \frac{H_Q^2}{2^A} + \frac{(R_Q+F_Q)^2}{B}$

The terms utilized in the theorem are H_Q, R_Q, F_Q, *A*, and *B* denotes the number of *Hash*, *Send*, *Execute* query,length of the hash function output value and range space of random number respectively.

Proof: We examine the security of the certificate generation similar to previous work of [34], [40], [41]. We also took the same assumption as taken by [38] during the query and game execution. α utilizes three games to get the secrets of the certificate generation process. The three games G_1, G_2, G_3 are used, and an event $Su_{\alpha G_1}$ could be explained asα predicts the random bit during the execution of the game,and $Pr[R_{\alpha G_1}]$ denotes the winning probability of G_1.

Game(G_1): α simulates this game to obtain the value of C in order to get the secrets before the Oracle finishes the initialization of the process.

$$Adv_\alpha = |2Pr\left[R_{\alpha G_1}\right] - 1] \tag{37}$$

Game (G_2): Now α intercepts the exchanged message betweenM^i, B^j and O_n^k by running the*Execute* query. After getting the exchanged message, α runs the*Test* query to get any secrets through which he can derive the private key of MNO. However, it is not confirmed by executing the query that the value is real and random. This means that α fails to predict the value and loss of the game since all the secrets are transferred in encrypted form using the digital signature. Therefore, the winning probability of G_1 and G_2 will be similar.

$$Pr\left[R_{\alpha G_2}\right] = Pr\left[R_{\alpha G_1}\right] \tag{38}$$

Game(G_3): In the game G_1 and G_2, α tries to get any insights by running the *Execute* and *Send* query, but he fails, now he will execute the other query in order to get the random number through the exchanged messages. Since all the random numbers used in certificate derivations are sent in signed messages and generated using elliptic curve cryptography. However, some random numbers are sent in plain text, but they will not be useful in determining the rest used in the certificate creations due to the discrete logarithm problem. Apart from that, it is observed that running the *Hash*query will not get any chance, and there will not be any collision. Thus, this shows that α also fails to win this game, and the winning probability of G_2 and G_3 are the same. Thus,we get the following outcome by adopting the birthday paradox.

$$Pr\left[R_{\alpha G_2}\right] - Pr\left[R_{\alpha G_3}\right] \leq \frac{H_Q^2}{2^{A+1}} + \frac{\left(R_Q + F_E\right)^2}{2B} \tag{39}$$

α has played the whole game and fails to win the game, so the winning probability to predict the bit C is

$$Pr\left[R_{\alpha G_3}\right] = \frac{1}{2} \tag{40}$$

from equations (37) (38), and (40), we can obtain

$$\begin{aligned} Adv_\alpha &= |2Pr\left[R_{\alpha G_1}\right] - 1| \\ \frac{1}{2}Adv_\alpha &= |Pr\left[R_{\alpha G_1}\right] - \frac{1}{2}| \\ &= Pr\left[R_{\alpha G_2}\right] - Pr\left[R_{\alpha G_3}\right] \end{aligned} \tag{41}$$

We get the following result from Eq. (39) and Eq. (41).

$$\frac{1}{2}Adv_{\alpha} \leq \frac{H_Q^2}{2^{A+1}} + \frac{\left(R_Q + F_Q\right)^2}{2B}$$
$$Adv_{\alpha} \leq \frac{H_Q^2}{2^{A}} + \frac{\left(R_Q + F_Q\right)^2}{B} \quad (42)$$

This demonstrates that the attackers will not get any insights during the certificate creation to obtain the secrets used within polynomial time.

C. Perfect Forward Secrecy

This theorem is used to show that both levels preserve the perfect forward secrecy at the time of the certificate creation.

Theorem 2: Let us assume that α tries to get secrets stored on M^i, B^j and O_n^k and can derive the secrets that are different for different certificate creations then $Adv_{\alpha} \leq \frac{H_Q^2}{2^A} + \frac{(R_Q+F_Q)^2}{B} + 2A_{A_d}^{ECDDH}$

The G_1, G_2, and G_3 will be expected to be similar to Theorem 1.

(G_4): This game is similar toG_3 except the execution of the *Corrupt* query. The α executes the *Corrupt* query to get the random number used in the private key derivation since these numbers are not exchanged in the public channel. ECC is applied, and random numbers used in the derivation are generated that will not allow the attacker to get these random numbers; still, they have private and public keys of communicating entities through which they exchange the messages. Due to the hardness of ECQV (see Section III-A). Thus, due to the use of ECC, it is impossible to determine the secrets used in certificate generation even if α has secrets stored on the communication entities. So we have

$$Pr\left[Su_{\alpha}G_4\right] - Pr\left[R_{\alpha}G_2\right] \leq \frac{H_Q^2}{2^{A+1}} + 2A_{\alpha}^{ECDDH} \quad (43)$$

All four games have been played by the attacker to predict the value of C and then

$$Pr\left[Su_{\alpha}G_4\right] = \frac{1}{2} \quad (44)$$

from equations (37) (38), and (44), we can obtain

$$Adv_{\alpha} = |2Pr[Su_{\alpha}G_1] - 1|$$
$$\frac{1}{2}Adv_{\alpha} = |Pr[Success_{\alpha}G_1] - \frac{1}{2}|$$
$$= Pr\left[Su_{\alpha}G_2\right] - Pr\left[Su_{A_d}G_4\right] \quad (45)$$

We obtain the following outcome from Eq. (43) and Eq. (45).

$$Adv_{\alpha} \leq \frac{H_Q^2}{2^{A}} + \frac{\left(R_Q + F_Q\right)^2}{B} + 2A_{\alpha}^{ECDDH} \quad (46)$$

Thus, the proof of Theorem 1 and 2 demonstrates that α can not obtain the secrets by intercepting the message and by getting the long-term secrets within polynomial time.

D. Formal Analysis Using Scyther Tool

The security of the certificate creation at MNO and NF levels is verified using the Scyther tool. This is a well-known tool used to validate security protocols [34], [42]. This tool utilizes the.spdl language to model the protocols. It uses

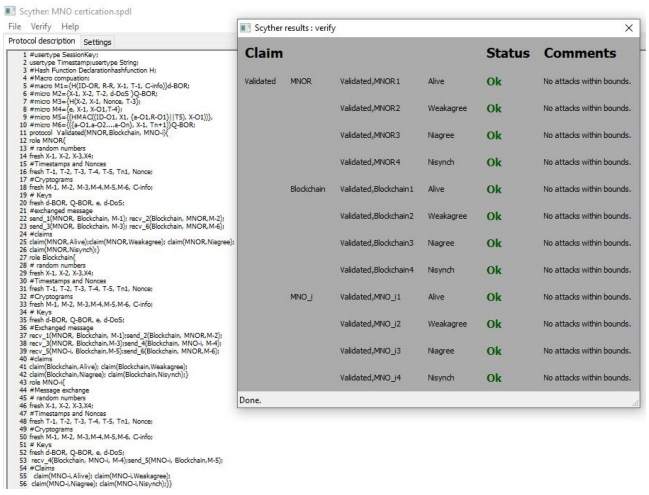

Fig. 21. Security verification for the MNO Certificate Issuance.

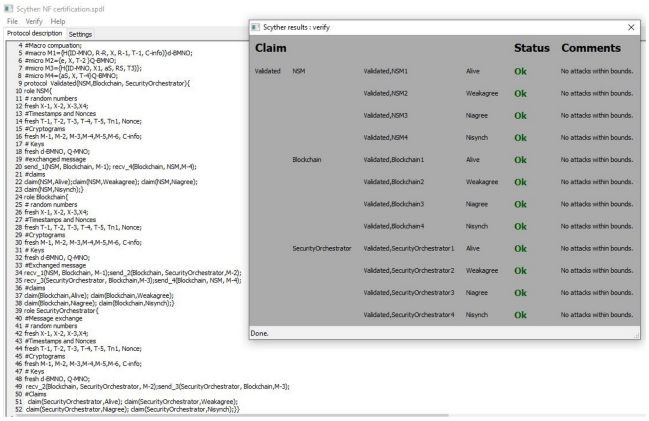

Fig. 22. Security verification for the NF Certificate Issuance.

four types of claims to verify the security of the proposed protocol under well-known threat models such as Dolev and CK adversary models. The description of the query is as follows [43]

- Alive- Successful execution of this claim indicates that all the events between the MNOR, Blockchain, MNO-i and NSM, Blockchain, and Security Orchestrator have been executed.
- Weakagree- Successful execution of this claim indicates that there is no DoS, impersonation, and replay attack.
- Nisynch- Successful execution of this claim indicates that the traceability attack is not possible during certificate creation.
- Niagree- Successful execution of this claim indicates that there is no non-repudiation attack and stolen verifier attack.

We execute both protocols (certificate creation at the MNO level and NF level), which shows that all the specified claims have been passed by the mechanism designed for certificate issuance at MNO and NF levels. Therefore, we can infer from the outcome of the Scyther tool shown in Fig. 21 and Fig. 22 that there is no attack during the certificate creation.

VI. Informal Analysis

This section offers informally demonstrative evidence that no attack can occur at the MNO and NF levels during certificate creation.

A. Resilient Against Replay Attack

In order to prevent the replay attack, both levels (i.e., MNO level and NF level) use timestamps and nonces in every exchanged message. When any entity receives the message, it verifies the freshness using $\Delta T \geq T_{i+1} - T_i$. We assume that the clock used by the communication entities during the certificate issuance is synchronized, similar to [41], [42]. If this condition matches, then it is believed that the message is fresh, and the process continues further. Otherwise, they reject the message.

B. Resilient Against Traceability Attack

If the attacker intercepts $<M1, M2, M3, M4>$ at NF level or $<M1, M2, M3, M4{:}1, ..M4{:}n, M6>$ at MNO level and tries to interlink the intercepted message then he/she can not succeed due to the use of random numbers that are different for a different session. In both levels, during the message exchange, random numbers are used to restrict the attacker from interlinking with each other by capturing the exchanged messages.

C. Resilient Against Impersonation Attack

If an attacker tries to forge the message $<M1, M2, M3, M4>$ at NF level or $<M1, M2, M3, M4{:}1, ..M4{:}n, M6>$ at MNO level in order to get the secrets through which he can determine the certificate key, then this is not possible. All the messages use the digital signature mechanism that restricts the attacker from forging the message. If the attacker tries, then he will fail due to the random numbers used, which are unique in the process.

D. Resilient Against Non-Repudiation

The messages exchanged at both levels are signed, and only the legitimate entity can decrypt this. Therefore, if an entity denies it, then others can prove that the message is signed by your private key, and only you can decrypt it.

E. Resilient Against DoS Attack

Both at the MNO and NF levels,random numbers and timestamps are used to make sure that entities involved in the communication will easily and rapidly identify that the message is replayed.

F. Perfect Forward Secrecy

If the attacker obtains the public and private keys of the entity involved at the MNO level and NF level before the starting of the certificate process, then he can not derive the $\{(a_K = rO{:}K + e * d_{OK}), (Q_{O:R} = e * P + R_{O:1} + \cdots + R_{O:n}), (d_{O:R} = e * r_R + a_{O:1} + \cdots + a_{O:n})\}$ due to the $\{e, r_R, r_{O:K}\}$ at MNO level and $\{(a_S = r_S + e * d_S), (Q_{NF} = e * P + R_S), (d_{NF} = e * r_R + a_S)\}$ due to $\{r_R, r_S\}$ that is used in the certificate derivation due to the hardness of the discrete logarithm problem (see Section III-A). Although the attacker has long-term secrets, he can not determine the random numbers due to the hardness of the discrete logarithm problem.

G. Stolen Verifier Attack

An attacker can not get the secrets used in certificate creation at both levels even if they acquire the table of MNO and MNR. All the secrets are derived using Elliptic Curve Qu-Vanstone (ECQV), and random numbers used in the process are not transmitted in the channel, which shows that the attacker can not take the random number of the MNO and MNR. This shows that an attacker having access to such a table will not be able to obtain secrets.

VII. Discussion

In this section, we discuss the limitations related to our solution and potential solutions for those limitations.

As we observed in the results, our framework takes a bit more time to generate certificates than the existing approaches. Currently, the smart contract communicates with the selected MNOs sequentially in our system. We can optimize it to communicate in a parallel way to reduce time consumption. In the current system, we add all the requests and responses related to the certificate generation into the ledger to increase accountability. This fills up the ledger rapidly. Optimized storage mechanisms can be developed for the ledger to address this challenge. Also, we can develop a system in which all the data is stored outside the ledger, but the hashes of the messages are stored in the ledger. Moreover, during the authorization process, our framework consumes more time than the existing mechanisms. A high-performance blockchain query mechanism can be implemented to reduce this time consumption.

VIII. Conclusion

This paper proposes a novel blockchain-based framework to eliminate the security challenges related to authentication and authorization in the 5G network-slicing ecosystem. A novel distributed multi-party certificate management framework using blockchain and ECQV certificates has been developed to facilitate the certificate management requirements to provide authentication services for network slicing in multi-operator environments. Also, we have designed the required secure communication protocols for certificate management. Moreover, the identified authorization security challenges in NF sharing between network slices are eliminated using the proposed framework. We implemented a prototype of the proposed framework using Hyperledger Fabric. The comprehensive experiments with the prototype showed that our framework could simplify the certificate management process in a multi-domain environment. For instance, it can eliminate interruptions in certificate generation and increase storage utilization efficiency. Also, the experiments showed that our authorization framework could mitigate the

security challenges in NF sharing between network slices. Formal and informal verification of communication protocols illustrated that our framework is secure considering the mentioned threat model.

In the current implementation, we considered only MNOs and NFs in the certificate generation process. In future works, we intend to optimize the certificate management framework for all related entities in the telecommunication domain, such as VNF providers and Infrastructure providers. Also, in the experimental results, we observed that our framework takes a bit longer than traditional approaches due to the sequential communication with the set of MNOs in the certificate generation. We will optimize the system to reduce the time for certificate generation in future works. Furthermore, we will consider integrating our system with existing certificate frameworks such as commercial CAs.

REFERENCES

[1] A. A. Abd El-Latif, B. Abd-El-Atty, W. Mazurczyk, C. Fung, and S. E. Venegas-Andraca, "Secure data encryption based on quantum walks for 5G Internet of Things scenario," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 1, pp. 118–131, Mar. 2020.

[2] H. Lian, Y. Yang, and Y. Zhao, "Efficient and strong symmetric password authenticated key exchange with identity privacy for IoT," *IEEE Internet Things J.*, vol. 10, no. 6, pp. 4725–4734, Mar. 2023.

[3] Q.-T. Luu, S. Kerboeuf, and M. Kieffer, "Uncertainty-aware resource provisioning for network slicing," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 79–93, Mar. 2021.

[4] S. Wijethilaka and M. Liyanage, "Survey on network slicing for Internet of Things realization in 5G networks," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 957–994, 2nd Quart., 2021.

[5] "Network slicing market, by component type (solution, services)." 2021. [Online]. Available: https://www.researchdive.com/5670/network-slicing-market

[6] R. F. Olimid and G. Nencioni, "5G network slicing: A security overview," *IEEE Access*, vol. 8, pp. 99999–100009, 2020.

[7] R. Chai, D. Xie, L. Luo, and Q. Chen, "Multi-objective optimization-based virtual network embedding algorithm for software-defined networking," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 1, pp. 532–546, Mar. 2020.

[8] "Security architecture and procedures for 5G system; (Release 17)," 3GPP, Sophia Antipolis, France, Rep. TS 33.501 V17.6.0, Jun. 2022.

[9] C. Jost and B. Smeets (Ericsson, Stockholm, Sweden). *Security for 5G Service-Based Architecture*. (Aug. 2020). [Online]. Available: https://www.ericsson.com/en/blog/2020/8/security-for-5g-service-based-architecture

[10] *A Slice in Time: Slicing Security in 5G Core Network*, AdaptiveMobile Secur., Dublin, Ireland, 2021.

[11] M. Li, L. Zhu, Z. Zhang, C. Lal, M. Conti, and M. Alazab, "Anonymous and verifiable reputation system for E-commerce platforms based on blockchain," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 4, pp. 4434–4449, Dec. 2021.

[12] J. Li, H. Yan, and Y. Zhang, "Certificateless public integrity checking of group shared data on cloud storage," *IEEE Trans. Services Comput.*, vol. 14, no. 1, pp. 71–81, Jan./Feb. 2018.

[13] V. A. Cunha et al., "Network slicing security: Challenges and directions," *Internet Technol. Lett.*, vol. 2, no. 5, p. e125, 2019.

[14] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, "Elliptic curve lightweight cryptography: A survey," *IEEE Access*, vol. 6, pp. 72514–72550, 2018.

[15] Website Builder Expert Staff. "How much does an SSL certificate cost in 2022?" Website Builder Expert. Aug. 2021. [Online]. Available: https://www.websitebuilderexpert.com/building-websites/ssl-certificate-cost/

[16] M. Campagna. "SEC 4: Elliptic curve Qu-Vanstone implicit certificate scheme (ECQV)." Standards Efficient Cryptography. 2013. [Online]. Available: https://www.secg.org/sec4-1.0.pdf

[17] G. O. Boateng, D. Ayepah-Mensah, D. M. Doe, A. Mohammed, G. Sun, and G. Liu, "Blockchain-enabled resource trading and deep reinforcement learning-based autonomous RAN slicing in 5G," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 1, pp. 216–227, Mar. 2022.

[18] A. M. Seid, A. Erbad, H. N. Abishu, A. Albaseer, M. Abdallah, and M. Guizani, "Blockchain-empowered resource allocation in multi-UAV-enabled 5G-RAN: A multi-agent deep reinforcement learning approach," *IEEE Trans. Cogn. Commun. Netw.*, vol. 9, no. 4, pp. 991–1011, Aug. 2023.

[19] J. Arkko (Ericsson, Stockholm, Sweden). *Service-Based Architecture in 5G*. (2017). [Online]. Available: https://www.ericsson.com/en/blog/2017/9/service-based-architecture-in-5g

[20] D. Boneh et al., "Multiparty non-interactive key exchange and more from isogenies on elliptic curves," *J. Math. Cryptol.*, vol. 14, no. 1, pp. 5–14, 2020.

[21] M. M. Payeras-Capellà, M. Mut-Puigserver, M. À. Cabot-Nadal, and L. Huguet-Rotger, "Blockchain-based confidential multiparty contract signing protocol without TTP using elliptic curve cryptography," *Comput. J.*, vol. 65, no. 10, pp. 2755–2768, 2022.

[22] C. Dai and Z. Xu, "A secure three-factor authentication scheme for multi-gateway wireless sensor networks based on elliptic curve cryptography," *Ad Hoc Netw.*, vol. 127, Mar. 2022, Art. no. 102768.

[23] M. Y. Kubilay, M. S. Kiraz, and H. A. Mantar, "CertLedger: A new PKI model with certificate transparency based on blockchain," *Comput. Secur.*, vol. 85, pp. 333–352, Aug. 2019.

[24] T. Hewa, A. Bracken, M. Ylianttila, and M. Liyanage, "Blockchain-based automated certificate revocation for 5G IoT," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–7.

[25] T. Saleem et al., "ProofChain: An X.509-compatible blockchain-based PKI framework with decentralized trust," *Comput. Netw.*, vol. 213, Aug. 2022, Art. no. 109069.

[26] A. Yakubov, W. Shbair, A. Wallbom, D. Sanda, and R. State, "A blockchain-based PKI management framework," in *Proc. 1st IEEE/IFIP Int. Workshop Manag. Manag. Blockchain (Man2Block) Colocat. IEEE/IFIP NOMS*, 2018, pp. 1–6.

[27] B. Khieu and M. Moh, "CBPKI: Cloud blockchain-based public key infrastructure," in *Proc. ACM Southeast Conf.*, 2019, pp. 58–63.

[28] Y. C. E. Adja, B. Hammi, A. Serhrouchni, and S. Zeadally, "A blockchain-based certificate revocation management and status verification system," *Comput. Secur.*, vol. 104, May 2021, Art. no. 102209.

[29] X. Luo, Z. Xu, K. Xue, Q. Jiang, R. Li, and D. Wei, "ScalaCert: Scalability-oriented PKI with redactable consortium blockchain enabled "on-cert" certificate revocation," in *Proc. IEEE 42nd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2022, pp. 1236–1246.

[30] W.-Y. Chiu, W. Meng, and C. D. Jensen, "ChainPKI—Towards Ethash-based decentralized PKI with privacy enhancement," in *Proc. IEEE Conf. Dependable Secure Comput. (DSC)*, 2021, pp. 1–8.

[31] J. Yan, X. Hang, B. Yang, L. Su, and S. He, "Blockchain based PKI and certificates management in mobile networks," in *Proc. IEEE 19th Int. Conf. Trust, Security Privacy Comput. Commun. (TrustCom)*, 2020, pp. 1764–1770.

[32] J. Yan, B. Yang, L. Su, S. He, and N. Dong, "Decentralized certificate management for network function virtualization (NFV) implementation in 5G networks," in *Proc. Int. Conf. Mobile Multimedia Commun.*, 2021, pp. 81–93.

[33] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[34] A. K. Yadav, M. Misra, P. K. Pandey, A. Braeken, and M. Liyange, "An improved and provably secure symmetric-key based 5G-AKA protocol," *Comput. Netw.*, vol. 218, Dec. 2022, Art. no. 109400.

[35] T. Liu, F. Wu, X. Li, and C. Chen, "A new authentication and key agreement protocol for 5G wireless networks," *Telecommun. Syst.*, vol. 78, pp. 1–13, Jul. 2021.

[36] J. Aas et al., "Let's encrypt: An automated certificate authority to encrypt the entire web," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2019, pp. 2473–2487.

[37] D. A. Ha, K. T. Nguyen, and J. K. Zao, "Efficient authentication of resource-constrained IoT devices based on ECQV implicit certificates and datagram transport layer security protocol," in *Proc. 7th Symp. Inf. Commun. Technol.*, 2016, pp. 173–179.

[38] M. Abdalla, P.-A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. Int. Workshop Public Key Cryptography*, 2005, pp. 65–84.

[39] C. J. F. Cremers, "Scyther: Semantics and verification of security protocols," Ph.D. dissertation, Dept. Math. Comput. Sci., Eindhoven Univ. Technol., Eindhoven, The Netherlands, 2006.

[40] M. Wazid, A. K. Das, N. Kumar, and M. Alazab, "Designing authenticated key management scheme in 6G-enabled network in a box deployed for industrial applications," *IEEE Trans. Ind. Informat.*, vol. 17, no. 10, pp. 7174–7184, Oct. 2021.

[41] S. Son, J. Lee, Y. Park, Y. Park, and A. K. Das, "Design of blockchain-based lightweight V2I handover authentication protocol for VANET," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 3, pp. 1346–1358, May/Jun. 2022.
[42] A. K. Yadav, M. Misra, P. K. Pandey, and M. Liyanage, "An EAP-based mutual authentication protocol for WLAN-connected IoT devices," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1343–1355, Feb. 2023.
[43] S. Shunmuganathan, "A reliable lightweight two factor mutual authenticated session key agreement protocol for multi-server environment," *Wireless Pers. Commun.*, vol. 121, no. 4, pp. 2789–2822, 2021.

Shalitha Wijethilaka (Student Member, IEEE) received the bachelor's degree (First Class Hons.) in electronics and telecommunication engineering from the University of Moratuwa, Sri Lanka, in 2017, and the Ph.D. degree from the School of Computer Science of University College Dublin, Ireland. He is currently a Data Engineer with Dun & Bradstreet. He has industrial experience in IoT and telecommunication networks from 2017 to 2020. Moreover, he works as a Teaching Assistant with UCD. His research mainly focuses on improving network slicing security in telecommunication networks. In addition, he is interested in blockchain and federated learning in telecommunication, metaverse realization, and IoT security. He works as a Reviewer of IEEE ACCESS, Spring Nature, and IEEE ICC. For more information, see https://ucdcs-research.ucd.ie/phd-student/shalitha-wijethilaka.

Awaneesh Kumar Yadav (Member, IEEE) received the Ph.D. degree from the Indian Institute of Technology Roorkee, India. He is currently a Postdoctoral Researcher with the School of Computer Science, University College Dublin, Ireland. His research area includes network security, 5G/6G security, IoT and cloud security, and post quantum security.

An Braeken (Senior Member, IEEE) received the degree in applied sciences from COSIC at the KULeuven in 2006. She is a full-time Professor with VUB and the Director of VUB/ULB Postgraduate Digital and IT Essentials. She is associated with the research group ETRO.RDI at VUB. She is the co-author of over 300 publications and has been managing more than 30 national and international projects. Her interests include lightweight security and privacy protocols for IoT, cloud and fog, blockchain, and 5G security.

Madhusanka Liyanage (Senior Member, IEEE) received the Doctor of Technology degree in communication engineering from the University of Oulu, Oulu, Finland, in 2016. He is an Associate Professor/Ad Astra Fellow and the Director of Graduate Research with the School of Computer Science, University College Dublin, Ireland. He is an Honorary Adjunct Professor with the University of Ruhuna, Sri Lanka, and with the University of Sri Jayawardhanepura, Sri Lanka. He was also a recipient of the prestigious Marie Skłodowska-Curie Actions Individual Fellowship and the Government of Ireland Postdoctoral Fellowship from 2018 to 2020. His research interests are 5G/6G, blockchain, network security, artificial intelligence, explainable AI, federated learning, network slicing, Internet of Things, and multiaccess edge computing. In 2020, he received the "2020 IEEE ComSoc Outstanding Young Researcher" Award by IEEE ComSoc EMEA. Also, he was awarded an Irish Research Council (IRC) Research Ally Prize as part of the IRC Researcher of the Year 2021 awards for the positive impact he has made as a supervisor. In 2022, he received "2022 The Tom Brazil Excellence in Research Award" by SFI CONNECT Center. In 2021, 2022, and 2023, he was ranked among the World's Top 2% Scientists (2020, 2021, and 2022) in the List prepared by Elsevier BV, Stanford University, USA. For more information, see www.madhusanka.com.