NetChain: A Blockchain-Enabled Privacy-Preserving Multi-Domain Network Slice Orchestration Architecture

Guobiao He, *Student Member, IEEE*, Wei Su, Shuai Gao, *Member, IEEE*,
Ningchun Liu, *Student Member, IEEE*, and Sajal K. Das, *Fellow, IEEE*

***Abstract*—Multi-domain networking slice orchestration is an essential technology for the programmable and cloud-native 5G network. However, existing research solutions are either based on the impractical assumption that operators will reveal all the private network information or time-consuming secure multi-party computation which is only applicable to limited computation scenarios. To provide agile and privacy-preserving end-to-end network slice orchestration services, this paper proposes NetChain, a multi-domain network slice orchestration architecture based on blockchain and trusted execution environment. Correspondingly, we design a novel consensus algorithm CoNet to ensure the strong security, scalability, and information consistency of NetChain. In addition, a bilateral evaluation mechanism based on game theory is proposed to guarantee fairness and Quality of Experience by suppressing the malicious behaviors during multi-domain network slice orchestration. Finally, the prototype of NetChain is implemented and evaluated on the Microsoft Azure Cloud with confidential computing. Experiment results show that NetChain has good performance and security under the premise of privacy-preserving.**

***Index Terms*—Multi-domain network slicing, privacy-preserving, blockchain, TEE.**

I. Introduction

THE 5TH Generation of Mobile Networks (5G) [1] revolutionizes the communication service experience and enables new applications in diverse domains such as Tactile [2] and Industry Internet [3]. 5G introduces the concept of network slice (NS), which is defined as an independent end-to-end (E2E) logical network running on a shared infrastructure (i.e., compute, storage, connectivity resources) capable of providing a negotiable service quality agreed among its consumers and providers. NS is based on NFV (Network Function Virtualization) and SDN (Software Defined Networking), enabling the E2E provisioning of network resources to meet vertical industries' service requirements [4], [5]. Emerging scenarios call for an agile end to end network slice orchestration across multiple administrative domains in 5G [6], [7]. However, this faces many challenges due to lacking trust such as privacy disclosure [8], [9], fairness, inconsistent billing, etc [10]. To cope with the challenges, various schemes are proposed, which can be classified as traditional and blockchain-based solutions.

Traditional multi-domain network slice architecture [11], [12], and [13] lacks privacy consideration. Besides, all these solutions rely on a centralized authority such as a broker to interchange complete network resource information. However, in a real environment, network operators are unwilling to disclose their private network resource information such as traffic matrices and prices, which may allow their competitors to estimate their future bidding prices. Moreover, the centralized authority faces single-point failure risk or may conduct malicious behaviors during the network slice orchestration. The work in [14], [15], and [16] are the traditional privacy-preserving solutions. However, these solutions are either based on weak privacy guarantee or time-consuming MPC (Secure Multi-party Computation) which is only applicable to limited computation scenarios. In addition, there are no effective mechanism to guarantee the fairness and suppress malicious behaviors during multi-domain slice orchestration since network operators may overstate their available network resource due to economic benefit.

Blockchain is an emerging technology and has been widely employed to enhance the trust in the distributed system, which has outstanding characters such as decentralization, tamper-proofing, high security, etc. The work in [17], and [18] are blockchain-based architectures for multi-domain slice orchestration. However, both of them need all the administrative domains to transparently share the network status information and lack privacy consideration. Moreover, both of them are based on existing third party blockchain platforms such as Ethereum [19] or Hyperledger Fabric [20], in which the underlying consensus algorithm faces security risks

Manuscript received February 8, 2021; revised June 23, 2021; accepted August 30, 2021. Date of publication September 3, 2021; date of current version March 11, 2022. This work was supported by the National Key Research and Development Program of China (No. 2019YFBI802503), and the project National Nature Science Foundation of China (No. 61972026). The work of Sajal K. Das is partially supported by the U.S. National Science Foundation (NSF) grants under award numbers SATC-2030624 and CNS-1818942. The associate editor coordinating the review of this article and approving it for publication was M. Tortonesi. *(Corresponding author: Shuai Gao.)*

Guobiao He, Wei Su, and Ningchun Liu are with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China (e-mail: 17111014@bjtu.edu.cn; wsu@bjtu.edu.cn; 16111014@bjtu.edu.cn).

Shuai Gao is with with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China, and also with the PCL Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen 518066, China (e-mail: shgao@bjtu.edu.cn).

Sajal K. Das is with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: sdas@mst.edu).

Digital Object Identifier 10.1109/TNSM.2021.3110057

or scalability challenges. The same as traditional solutions, there is no effective mechanism to ensure QoE (Quality of Experience)/QoS (Quality of Service) and fairness during multi-domain slice orchestration in the current blockchain-based architectures.

To cope with the security risks and challenges of current solutions in multi-domain network slice orchestration, this paper presents a privacy-preserving architecture called NetChain based on blockchain and trusted execution environment (TEE). The design of NetChain is based on the key observation that blockchain and TEE have complementary properties [21]. Blockchain can enhance the trust between multi-domain operators and the architecture cannot be controlled by a single authority. TEE is adopted to guarantee information privacy with high computing performance. To ensure strong consistency, scalability, and security of NetChain we propose a novel consensus algorithm called CoNet, minimizing the performance impact of a blockchain-enabled architecture on multi-domain network slice orchestration. Moreover, a bilateral evaluation mechanism based on game theory is presented to ensure QoE/QoS and fairness by suppressing malicious behaviors in the system, inspiring the network operators to share their available network resources. In summary, this paper makes the following contributions.

- A blockchain-enabled privacy-preserving multi-domain network slice architecture called NetChain is presented, eliminating privacy leak, security risks, and poor QoE/QoS guarantee in the current solutions.
- A novel consensus algorithm is designed to guarantee the strong consistency, scalability, and security of NetChain.
- A bilateral evaluation mechanism based on game theory is proposed to ensure QoS/QoE and fairness during multi-domain slice orchestration.
- NetChain is implemented and evaluated on Microsoft Azure Cloud with confidential computing. The experiment results show that NetChain performs well in privacy, security, and performance.

The structure of the paper is organized as follows. Section II reviews the related work. Section III introduces the proposed NetChain architecture. The technical details of our novel consensus algorithm are illustrated in Section IV. In Section V, a bilateral evaluation mechanism based on game theory is proposed to ensure QoE/QoS and fairness by suppressing malicious behaviors during multi-domain network slice orchestration. The implementation and experiment results are given in Section VI. Section VII discusses the advantages, limitations and compatibility of NetChain. Finally, the conclusion and future works are presented in Section VIII.

II. Related Work

This section reviews the related work to the multi-domain network slice orchestration, which can be classified as traditional and blockchain-based solutions.

A. Traditional Solutions

The work in [11] is a centralized multi-domain network slicing orchestration architecture. However, it lacks privacy consideration, which makes the system difficult to deploy in a real network [22]. Moreover, the centralized architecture faces single-point failure risk. The centralized service broker may conduct malicious behaviors to influence the fairness during multi-domain network slice orchestration. Mano *et al.* [14] employs secure multi-party computation (MPC) for masking sensitive values during multi-domain slice orchestration. However, it only focuses on minimizing inter-infrastructure providers' virtual network prices. Multi-domain network slice orchestration factors are not only price, but also others such as bandwidth, delay, etc. The general-purpose MPC solutions are still complex and time-consuming. Francescon *et al.* [15] presents a cross-domain service orchestration framework X-MANO. It introduces an information model enabling each domain to advertise in a confidentially preserving network information to an external entity such as Federation Manager [15]. However, X-MANO can only provide a weak privacy guarantee since an external entity may be malicious and violate the privacy promise. In addition, it relies on a centralized Federation Manager to conduct cross-domain orchestration, which faces of single-point failure risk.

Joshi and Kataoka [16] uses a multi-domain orchestrator to deploy the SFC (Service Function Chain) across domains. It is a centralized architecture, which faces single-point failure risks. Moreover, the lack of detailed resource information about every domain makes the multi-domain SFC orchestration process more complicated and time-consuming. Finally, there is no effective mechanism to cope with the malicious behaviors, which may result in poor QoE/QoS.

B. Blockchain-Based Solutions

Rosa and Rothenberg [17] presents blockchain-based decentralized applications for multiple administrative domain networking. However, it needs all the administrative domains to transparently share the network status information, which lacks privacy consideration. In a real environment, the network operators are unwilling to disclose their private network information such as traffic matrices and prices, which may allow their competitors to estimate their future bidding prices. Besides, this approach is implemented on a third-party platform Ethereum [19], the network resource sharing and slice orchestration transaction processing performance may be greatly influenced by the network situation of Ethereum, resulting in poor QoS/QoE. For example, in 2017, the popularity of the CryptoKitties game in Ethereum cause the network to become heavily congested, slowing transaction processing significantly.

Afraz and Ruffini [18] proposes the blockchain-based 5G network slice brokering market. However, it also lacks privacy consideration. Besides, it is deployed on a third-party platform Hyperledger Fabric [20] using Raft [23]. The Raft consensus algorithm has poor scalability since the communication overhead increase exponentially with the increase of network size. Therefore, the performance will decrease sharply when the network size is large. The typical application scenarios of Raft are only limited to dozens of nodes. Besides, the leader selection process is fixed and can be predicted in Raft, which makes it vulnerable to DoS attacks.

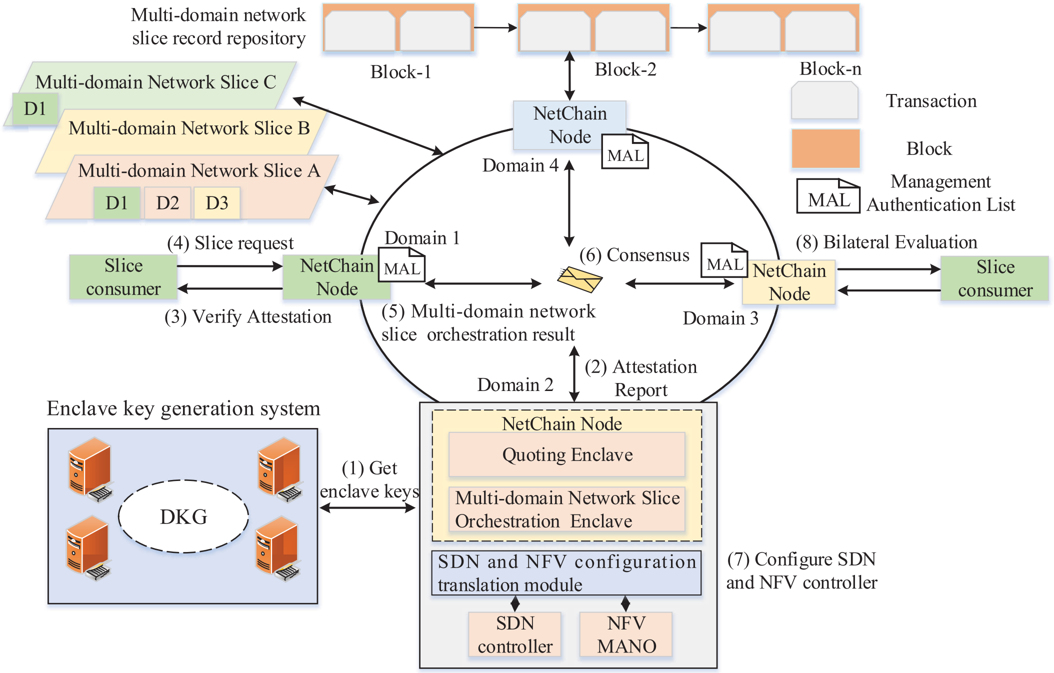
Multi-domain network slice record repository
Block-1
Block-2
Block-n
Transaction
Block
MAL
Management Authentication List
Multi-domain Network Slice C
D1
Multi-domain Network Slice B
Multi-domain Network Slice A
D1
D2
D3
NetChain Node
MAL
Domain 4
Domain 1
(4) Slice request
Slice consumer
NetChain Node
MAL
(3) Verify Attestation
(6) Consensus
MAL
NetChain Node
(8) Bilateral Evaluation
Slice consumer
(5) Multi-domain network slice orchestration result
Domain 3
Domain 2
(2) Attestation Report
Enclave key generation system
DKG
(1) Get enclave keys
NetChain Node
Quoting Enclave
Multi-domain Network Slice Orchestration Enclave
SDN and NFV configuration translation module
(7) Configure SDN and NFV controller
SDN controller
NFV MANO

Fig. 1. NetChain architecture.

III. NetChain Architecture

A. Requirements in Multi-Domain Slice Orchestration

We fully analyze the requirements in multi-domain slice orchestration as follows.

- *Privacy-Preserving:* Privacy-preserving is necessary to prevent private information disclosure during multi-domain slice orchestration.
- *Full Decentralization:* To eliminate the security risks in centralized architecture such as single point failure, manipulation attack, etc, multi-domains slice orchestration architecture must be full decentralization.
- *Tamper-Proofing and Consistency:* The multi-domain slice orchestration information should be tamper-proofing and consistency.
- *Fairness and QoE/QoS:* It is critical to guarantee the fairness and QoE/QoS in multi-domain slice orchestration by suppressing malicious behaviors in the system.
- *High Security:* It is important to design a multi-domains slice orchestration architecture that have inherent ability to mitigate DoS attacks, Sybil attacks, etc.
- *High Scalability:* High scalability is a crucial factor to aggregate more network administrative domains to share network resources.
- *Fast Provision:* The multi-domain slice orchestration process must be agile to meet the latency requirements of the emerging applications.

B. Security Threats in Multi-Domain Slice Orchestration

NetChain is a privacy-preserving multi-domain slice orchestration architecture based on blockchain and TEE, which must also protect from attacking to NetChain itself, threatening the security of multi-domain slice orchestration. Security threats that need to be defended are as follows.

- *Sybil Attacks:* An adversary creates many pseudonyms to disrupt the network and influence the system security.
- *Network Partition Attacks:* In a network partition attack, an attacker can isolate a set of nodes to impede the consensus or intercept network traffic. In NetChain, we need to ensure that the system cannot be controlled by the adversary even under network partition attacks.
- *DoS Attacks:* DoS attacks are typically accomplished by flooding the targeted server or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled.
- *Malicious Attacks:* An attacker in control of a fraction of nodes may conduct malicious attacks such as tamper-proofing, replay attacks, etc.

C. Overview of NetChain

To meet the requirements and cope with security threats during multi-domain network slice orchestration, NetChain introduces following techniques. A NetChain node is added in each network administrative domain to in charge of network slice orchestration. Then, a management authentication list maintains a white list of network operators information to prevent Sybil attacks. Third, NetChain adopts TEE to guarantee the information privacy and a distributed enclave key generation system to produce enclave keys. To enhance the security, scalability and consistency of the architecture, we design a novel consensus algorithm CoNet. To guarantee fairness and QoE/QoS, a bilateral evaluation mechanism based on game theory is presented. Finally, the SDN and NFV configuration translation module is added to translate the output of multi-domain slice orchestration results into executable SDN and NFV configuration.

The workflow of multi-domain network slice orchestration in NetChain is illustrated in Fig. 1.

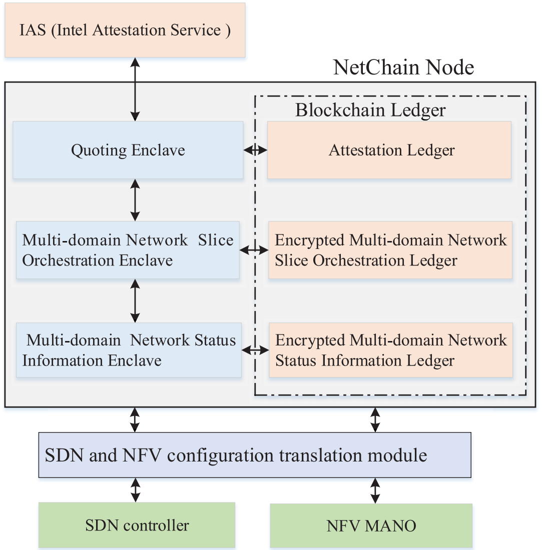
IAS (Intel Attestation Service)
NetChain Node
Blockchain Ledger
Quoting Enclave
Attestation Ledger
Multi-domain Network Slice Orchestration Enclave
Encrypted Multi-domain Network Slice Orchestration Ledger
Multi-domain Network Status Information Enclave
Encrypted Multi-domain Network Status Information Ledger
SDN and NFV configuration translation module
SDN controller
NFV MANO

Fig. 2. NetChain Node Components.

(1) First, each application enclave in NetChain node gets a public and private key pair from the enclave key generation system during initiation.

(2) Then, the enclave will then create an attestation report for the initial state of multi-domain network slice orchestration enclave using Intel Attestation Service (IAS) [24].

(3) Before sending a multi-domain network slice request, the slice consumer will verify the remote attestation report to make sure that the enclave environment and all the application data inside the enclave are safe and trustworthy.

(4) If the remote attestation succeeds, the slice consumer then sends the multi-domain network slice orchestration request to the corresponding NetChain node based on a secure Transport Layer Security (TLS) channel.

(5) After receiving the request, the NetChain node conducts network slice orchestration based on the decrypted multi-domain network resource status information. The encrypted network slice orchestration result is broadcast to all NetChain nodes for consensus.

(6) After the consensus, the slice orchestration result (i.e., required bandwidth, storage, and computing capacity in each domain) will be stored in the blockchain with the encryption format, which can only be decrypted inside the enclave.

(7) The output of slice orchestration results are translated into the executable SDN and NFV configuration file by translation module. The NFV Management and orchestration (MANO) and SDN controller will establish a multi-domain network slice based on corresponding configuaration file.

(8) After the end of the multi-domain network slicing service, NetChain will start bilateral evaluation in Section V to guarantee fairness and QoE/QoS of the service.

D. NetChain Node Components

NetChain node is the most critical entity in NetChain, which is in charge of multi-domain network slice orchestration. NetChain node components are illustrated in Fig. 2, including a quoting enclave, multi-domain network slice orchestration enclave, multi-domain status information enclave, and their corresponding ledgers stored in blockchain.

Quoting Enclave: Quoting Enclave is used to generate an attestation quote for application enclave, proving to a remote party that application software inside the enclave is intact and trustworthy. After enclave's initialization, it generates a new public-private key-pair within the enclave and creates an attestation report that summarizes enclave and platform state. The attestation report includes an SHA-256 hash of the entire application code and the corresponding library that is supposed to run in the enclave, which is sent to the quoting enclave. The Quoting Enclave authenticates the report and converts the body of the attestation report into a quote and signs it with the Intel Enhanced Privacy ID (Intel EPID) key provided by Intel Attestation Service (IAS) [24]. This quote indicates that application enclave is indeed a genuine and trustworthy SGX enclave running the code it claims, which is passed to remote entities off the platform for verification.

Multi-Domain Network Slice Orchestration Enclave: The multi-domain network slice orchestration application and its corresponding code library are stored in the enclave, executing multi-domain network slice orchestration based on the latest network status information. Any change of the multi-domain network slice orchestration application and its corresponding code library needs the consensus of majority administrative network providers. The network slice orchestration results containing slice owner, life-circle, and the corresponding network resource needed in each administrative domain are encrypted and broadcast to the network for consensus. After the consensus, the encrypted network slice orchestration results are stored in blockchain. Then, the slice orchestration results are translated into the executable SDN and NFV configuration file by translation module in each administrative domain. Finally, SDN controller and NFV MANO will deploy the network resource and NFs in each administrative domain according to the corresponding configuration file.

Multi-Domain Network Status Information Enclave: Multi-domain network status information enclave maintains a latest and consistent privacy-preserving network status information repository. The network status information within the enclave will be updated in each round after the consensus. The multi-domain network status information enclave exposes an interface to the multi-domain network slice orchestration enclave for accessing the latest network status information, guaranteeing the consistency of network status information and fairness during slice orchestration.

Blockchain Ledger: There are three types of information stored in the blockchain, including the attestation, encrypted multi-domain network slice orchestration information, and encrypted network resource status information. The attestation information is stored in the blockchain, proving to a local or remote party that the application of multi-domain network slice orchestration inside the enclave is intact and trustworthy. As the attestation information is transparent and tamper-proofing, the remote verifier can make an informed trust decision about the behavior of the network slice orchestration application inside the attested enclave. Multi-domain network slice orchestration information such as the owner of the network slice and the involved network resource information will be encrypted and stored in the blockchain, which could be decrypted in the network slice orchestration enclave. The encrypted network resource status information (e.g., the available storage, computing, bandwidth capacity, network resource price) will be broadcast to the network and stored in the blockchain, which can be decrypted in the multi-domain network status information enclave.

E. Enclave Key Generation System

The enclave keys are critical to guarantee the privacy of the multi-domain orchestration and network resource status information. However, the TEE enclave faces several security risks such as side-channel attacks. Once the TEE enclave of a NetChain node is compromised, all the previously confidential information will be leaked. In addition, if the enclave keys are lost, the data inside it can never be decrypted.

To eliminate the risks, Distributed Key Generation (DKG) [25] protocol is adopted in NetChain to generate application enclave keys. The DKG protocol allows a set of *n* nodes to collectively generate a secret with its shares spread over the nodes such that any subset of size greater than a threshold *t* can reveal or use the shared secret, while smaller subsets do not have any knowledge about it. DKG is byzantine-fault tolerant and confidential preserving. Moreover, the DKG committee members can be dynamically expanded without changing the secret by using proactive secret sharing [26].

To get the keys from DKG, a NetChain node establishes secure channels and authenticates itself with the DKG members. After verification, NetChain node collects *t* outcomes from DKG, constructing the secret and public key. The public keys of all enclaves will be broadcast to the network, encrypting the data across enclaves. TEE enclave key generation system could run in the TEE environment, which follows the policy that only the application enclave can get the secret shares of DKG and the members of DKG cannot get the secret shares of others. The application enclave will conduct remote attestation before collecting the secret shares form DKG, ensuring that the running environment of DKG is safe.

F. Management and Authentication List

Without the access control mechanism, the malicious network resource providers may launch Sybil attacks by creating a large number of pseudonymous identities to threaten consensus security. To prevent this from happening, NetChain introduces the management and authentication list as an access control mechanism, which contains a white list of network resource provider information.

The formation of the management authentication list is as follows. First, each NetChain node produces a pair of private and public key. The multi-domain network slice orchestration enclave and network status enclave get a pair of public and private key from the enclave key generation system. Then, all the public keys as unique identities for authentication are broadcast to the network. The format of the management certification list is defined as follows, $M = [N_{id}, P_k, E^s_{pk}, E^n_{pk}, H_m]$. Here, N_{id} is the identity of network resource providers, and P_k is the corresponding public key. E^s_{pk} and E^n_{pk} are the public key of slice orchestration enclave and network status enclave respectively. H_m is the hash of the management authentication list. Finally, the management and authentication list will be stored in the blockchain, which will be updated when the corresponding information is changed. Each network resource provider maintains a consistent view of the management and authentication list to prevent malicious nodes from joining the system.

G. Transaction Format of NetChain

There are three types of blockchain ledger in a NetChain node. Correspondingly, there are also three types of transactions, including network slice transaction, network status transaction, and attestation transaction.

Multi-Domain Network Slice Transaction: Multi-domain network slice transaction records the corresponding orchestration information, and its format is designed as follows.

SIG_i [Trans-type, PK_{slice}, Life-circle, NFs, Bill, SLAs, QoS/QoE, Trans-fee, Timestamp].

Three types of transactions for multi-domain network slice are identified by *Trans-type*, including creation, update, and revocation. PK_{slice} is the public key of a network slice consumer. *Life-circle* is the life circle of multi-domain network slice. *NFs* are the specific network functions of a network slice. The *Bill* is the billing information of the corresponding NFs cost. The *SLA* (Service-Level Agreement) defines the contract between a slice consumer and a network service provider, including the requirements of latency, bandwidth, price agreement, etc. *QoS/QoE* is the requirements of Quality of Service or Quality of Experience, including throughput, latency, frame loss ratio, etc. *Trans-fee* is added as the cost of network slice operation, which is an incentive mechanism to inspire multi-domain network resource provider to deploy the NetChain and can also mitigate the malicious network slice operation attacks. The accurate generation time of transactions is recorded by *Timestamp* to prevent replay attacks.

Network Status Transaction: Network status transaction is used to synchronize network status information such as available bandwidth, latency so that the multi-domain network slice is orchestrated based on the latest network status transaction.

SIG_i [PK_{pro}, D-ID, Adj-ID, Ingress P-ID, Egress P-ID, B_a, Hop-latency, C_a, S_a, Timestamp].

PK_{pro} is the public key of a network service provider. *D-ID* and *Adj-ID* are the network device identity and the adjacent network device identity, which can be a switch, router, or server. *Ingress P-ID* and *Egress P-ID* are the ingress and egress port identity in a network device. B_a is the available bandwidth capacity in a network link. *Hop-latency* is the packet forwarding latency in a switch or router. For server, C_a is the available computation capacity and S_a is the available storage capacity. *Timestamp* records the accurate time of network status transaction.

Attestation Transaction: The attestation transaction contains the quote information for remote attestation, which is stored in the blockchain to prevent any single party such as IAS maliciously from changing the attestation information. Currently, attestation transaction adopts Intel SGX quote format.

SIG_{EPID}[MRENCLAVE, MRSIGNER, ISVPRODID, ISVSVN, ATTRIBUTES, User-data].

When the enclave code/data pages are placed inside the Enclave Page Cache (EPC), the CPU calculates the enclave measurement. It is a 256-bit hash that identifies the code and initial data to be placed inside the enclave, storing this value in the *MRENCLAVE* register. After an enclave successfully initialized, the CPU records a hash of the enclave author's public key in the *MRSIGNER* register, which is the identity of enclave author. The enclave author also assigns a *Product ID* (*ISVPRODID*) to each enclave, allowing the enclave author to segment enclaves with the same enclave author identity. Besides, the enclave author assigns a *Security Version Number* (*ISVSVN*) to each version of an enclave, which reflects the security property level of the enclave, and should monotonically increase with improvements of the security property. *ATTRIBUTES* illustrates the enclave mode such as debug mode. *User data* allows to establish a secure channel bound to the remote attestation process so that a remote server may provision secrets to the entity that has been attested.

H. Block Format of NetChain

In NetChain, the multi-domain orchestration transaction repository is stored in blockchain in the form of blocks. Each block contains a cryptographic hash of the previous block, a timestamp, and network slice orchestration information as transaction data. The block formats is shown as follows.

$B_r = SIG_i$ [*Type, r, V*, R_i, P_i^c, Q_{r-1}, $H(B_{r-1})$, M_e, S_h^a, S_h^s, S_h^n, M_h, T_s, T_x].

In NetChain, there are three types of blocks identified by *Type*, including the attestation, network slice orchestration, and network status information block. Here, *r* is the round number since the consensus algorithm runs in rounds. To ensure the evolvability of the block format, *V* is used to distinguish the version of block format. We introduce R_i as the network resource contribution value and P_i^c as the credibility of network resource provider *i* to quantify its contribution. In every round, a certain number of NetChain nodes are randomly selected to participate in the consensus process based on a random seed Q_{r-1}. Each block contains the hash $H(B_{r-1})$ of the previous block, in which Merkle root hash M_e of transactions is adopted to guarantee the data integrity. In NetChain, the latest status of information of blockchain ledger is cached in the NetChain node to guarantee high accessing performance. Therefore, the hash S_h^a, S_h^s, S_h^n, and M_h are used to ensure latest state integrity of the attestation, slice orchestration, network status, and management certification list information in the blockchain. The accurate generation time of a block is recorded by a timestamp T_s to prevent replay attacks. All the multi-domain slice orchestration operations such as creation, update, and revocation are recorded in blockchain as transactions T_x.

IV. Design of Consensus Algorithm

The consensus algorithm is critical for the performance and security of a blockchain-based system, which can be divided into four types: PoW, PoS, Practical Byzantine Fault Tolerant (PBFT), and hybrid consensus. We need to fully analyze whether or not these consensus algorithms can be directly used in a multi-domain network slice orchestration architecture.

PoW and PoS are commonly adopted in open blockchain systems such as Ethereum and Bitcoin. PoS and PoW both work in a fintech setup and have high security. Both of them can mitigate Sybil attacks based on computing power or money (stake) respectively, it is extremely hard for the attacker to gather a large portion of computing power or tokens in the system to launch Sybil attacks. However, PoW and PoS run in open systems, an attacker can maliciously create a large number of Sybil nodes with new node IDs to disrupt the network, threatening system security. Besides, The PoS needs all the participants to deposit a lot of stakes (money), and PoW will consume a lot of electricity and computing power. Hence, a blockchain-based multi-domain network slice orchestration architecture based on PoW or PoS is costly, making it difficult to deploy in a real environment. Last but not least, PoW and PoS face forking risks, which may lead to data inconsistency and security risks [27].

PBFT and its variants are most commonly used in permissioned blockchain systems such as Honey Badger [28] and Hyperledger [20]. However, they have poor scalability and face security risks. The performance of PBFT and its variant decreases exponentially as the number of consensus nodes increases. The typical application scenarios of PBFT are limited to a few or no more than dozens of nodes. Besides, the leader selection process in PBFT can be predicted, which is vulnerable to DoS attacks.

Elastico [29], ByzCoin [30], Omnileder [31], and Algorand [32] are the latest hybrid consensus algorithm. However, Elastico relies on PoW and faces performance challenges [29]. Byzcoin uses Schnorr collective signature to improve consensus performance, which has a high fail probability during consensus [30], [33]. Algorand is designed for cryptocurrency and its security is based on easily manipulated money. Omniledger is also designed for cryptocurrency and uses different shardings to store different blocks, while each node in multi-domain network resource sharing and slice orchestration architecture needs to maintain a full view of the whole transaction repository. Thus, it is not suitable for multi-domain network slice orchestration architecture.

In conclusion, the current consensus algorithms face security risks or scalability challenges and cannot be directly adopted in a blockchain-enabled multi-domain network slice orchestration architecture. Therefore, we need to design a novel consensus algorithm, minimizing the performance impact of blockchain on multi-domain network slice orchestration and enhance the security of our proposed architecture.

A. Overview of Consensus Algorithm CoNet

The design of CoNet is inspired by the consensus algorithm of PBFT, [34] and [35], in which the leader is randomly selected based on credence value and a penalty mechanism is adopted to suppress malicious behaviors. We do some revision according to the specific requirements of multi-domain network slice orchestration scenario. First, we replace the credence value with network resource contribution value to inspire all the network operators to share their resource since more network resource contribution value indicates more economic benefit. The credibility value between 0 and 1 is proposed to quantify the trusted network resource capacity since network resource providers may overstate its available network resource capacity. Then, we delete the penalty mechanism to improve consensus performance to meet agile requirements of multi-domain slice orchestration. Correspondingly, a bilateral evaluation mechanism is designed in Section V to guarantee the fairness and QoE by constraining the malicious behaviors in the system. Finally, we continue to adopt BLS [36] collective signature to improve the consensus performance, while leader in BLS is randomly selected based on trusted network resource contribution value to enhance security.

The proposed consensus algorithm CoNet is derived from PBFT and the consensus procedure is very similar to it. The only differences contain the following aspects. CoNet introduces the BLS signature to improve the consensus performance and a certain number of participants, not all of them, to participate in consensus to improve scalability. Besides, the leader and consensus members in CoNet are randomly selected rather than fixed, which cannot be predicted in advance to enhance consensus security. PBFT is a typical asynchronous consensus that works in weak synchrony assumption, i.e., a message can be received within a known maximum delay. PBFT can convergence in a known maximum delay as long as there are more than $2f + 1$ honest nodes in the system (f is the number of malicious nodes), and the proof is shown in [37]. Therefore, CoNet can also converge in a known maximum delay and the proof of convergence is the same as PBFT.

B. Design of CoNet

1) Consensus Member Selection: The CoNet executes in rounds. In each round, a certain number of NetChain nodes called consensus members will be randomly selected to participate in the consensus process to improve scalability and security. The more consensus members, the safer the system. Meanwhile, consensus performance such as latency and throughput will decline as more consensus members are added. Therefore, the number of consensus members needs make a balance between security and performance according to the requirements of the multi-domain slice orchestration scenario.

At the beginning of each round, each NetChain node runs the function (1) to determine whether it is selected as a consensus member or not.

$$.H(SIG_i(r, R_i, P_i^c, Q_{r-1}, H(B_{r-1}))) \leq p_c, \tag{1}$$

where

$$p_c = \frac{N_c}{n}. \tag{2}$$

Here, r is the round number since the consensus executes in rounds. R_i and P_i^c are the network resource contribution value and the credibility of the NetChain node i. Q_{r-1} is the random seed in the last round, guaranteeing the randomness of consensus member selection process in order to prevent DoS attacks. The probability p_c of a NetChain node being selected as a consensus member is the same, as illustrated in (2). N_c is the number of consensus members and n is the total number of NetChain nodes.

2) Leader Selection and Block Proposal: In blockchain, the node must synchronize the data from a unique data source to ensure consistency. Leader selection is used to select a unique data source, so that other nodes synchronize information from it. Therefore, leader selection is critical in the consensus process, which largely determinines the security and performance.

$$.H(SIG_i(r, R_i, P_i^c, Q_{r-1}, H(B_{r-1}))) \leq p^l, \tag{3}$$

where

$$p^l = \frac{R_i^c}{S_R}, \tag{4}$$

$$S_R = \sum_{i=1}^{n} R_i^c, \tag{5}$$

$$\mathrm{R}_i^c = R_i * P_i^c, \tag{6}$$

$$\mathrm{R}_i = p_c * C_i^a + p_s * S_i^a + p_l * L_i^a. \tag{7}$$

In CoNet, the leader is randomly selected based on (3). The probability of a NetChain node being selected as a leader is decided by trusted network resource contribution value R_i^c and the overall trusted network resource contribution value S_R, as is illustrated in (4). The higher the trusted network resource contribution value, the higher the probability of being selected as the leader. In a real network environment, the network resource contribution value will normally be decentralized when there are many network providers sharing their available network resources. Even if the resource contribution value is centralized, it can only influence the economic benefit of the provider and does not affect system security. The system security is still decided by the majority consensus of all network resource providers, not by the resource contribution value. There may be several NetChain nodes selected as leaders simultaneously, the one with the smallest hash will be the leader. Here, the credibility of a network resource provider P_i^c is introduced to quantify the credibility of corresponding network resource providers since they may overstate their available network resource. P_i^c is defined in Section V. As is shown in (7), the R_i consists of available computing resource C_i^a, available storage resource S_i^a, and available link bandwith L_i^a. Their corresponding price p_c, p_s, and p_l are used to quantify the network resource value.

The leader assembles multi-domain slice orchestration transactions into a block and produce a corresponding BLS collective signature. Consensus members will make consensus about the proposed block with a BLS collective signature. After consensus, it will be stored in blockchain.

$$Q_r = \mathrm{H}\big(SIG_i\big(Q_{r-1,}r\big)\big). \tag{8}$$

Algorithm 1 CoNet Consensus Algorithm

```
Initialize: r = 1, $Q_r$ = random number;
while TRUE do
    if $clock = 0$ then
        $C_{member} \leftarrow Consensus\_Member\_Sortition()$
        $C_{leader} \leftarrow Consensus\_Leader\_Sortition()$
        $C_{leader}$ proposes a block
    end if
    if clock = (0, 2$\lambda$) then
        $C_{member}$ verify the block
        if Verification = True then
            return Verification vote to the $C_{leader}$
            $C_{leader}$ produces a BLS signature of verification votes $\leftarrow$ BLSsign();
            $Q_r = H(SIG_i(Q_{r-1}, r))$
        else
            Leader_reselection vote
            Broadcast()
        end if
    end if
    if clock = (2$\lambda$, 4$\lambda$) then
        $C_{member}$ verify the collective signature of verification vote
        if Verification = True then
            return next-round vote to the $C_{leader}$
            $C_{leader}$ produces BLS signature of next-round votes $\leftarrow$ BLSsign();
        else
            Leader re-selection vote
            Broadcast()
        end if
        if received $\geq 2f + 1$ leader_reselection votes then
            return to clock = 0 and reselect a new leader
        end if
    end if
    if clock $\geq$ 4$\lambda$ then
        $C_{member}$ verify the BLS signature of next round votes
        if Verification = True then
            clock = 0
            r = r + 1
        else
            Leader re-selection vote
            Broadcast()
        end if
    else
        if received $\geq$ 2f + 1 leader reselection votes then
            return to clock = 0 and reselect a new leader
        end if
    end if
end while
```

In each round, the random seed Q_r will be renewed based on the signature of the consensus leader and the random seed in the last round. The leader and consensus members are randomly selected and changed based on random seed every round, which cannot be predicted in advance to ensure system security.

C. Workflow of CoNet

The consensus is used to guarantee the consistency of the multi-domain slice information repository. The consensus in NetChain executes in rounds based on period λ, guaranteeing that all the transactions and blocks will be received within a period of time. If λ is small, many NetChain nodes may not yet receive the message and result in consensus failure. Otherwise, the consensus latency will be long and result in poor performance. Therefore, λ should be carefully determined according to the network size. Each NetChain node keeps a timer which resets to 0 when a new round begins. The workflow of consensus is illustrated in Algorithm 1.

Initialization: When NetChain starts for the first time, the round number are initialized to 1. Q_r is a rand number known to all NetChain nodes. The random seed can be generated from a trusted randomness beacon or a distributed randomness generation protocol such as RandHound [38]. We assume that the multi-domain slice orchestration transactions have been broadcast to the network in advance. f is the maximum number of malicious nodes that the system can tolerate.

(1) *Clock = 0:* In each NetChain node, the clock will be reset to 0 after received a BLS collective signature of the next-round votes, which indicates that new round begins. Each NetChain node will run the hash sortition function to randomly select the consensus members and leader.

(2) *Clock = (0, 2λ):* The NetChain node being selected as the leader assembles all transactions into blocks and broadcast them to the network. All the consensus members will send a verification proof with signature to the leader after verified the proposed blocks. If there is no block proposed or the leader proposes a malicious block, the verification is false. Then, consensus members will produce a leader reselection vote, which is similar to the view change in PBFT.

(3) *Clock = (2λ, 4λ):* After receiving more than $\geq 2f + 1$ consensus members' verification proof with signature, the leader will produce a BLS collective signature for the proposed block and broadcast it to the network for final confirmation. The consensus members will verify the corretness of BLS collective signature of verification votes and produces next-round vote to the leader after verification. If the verification is false, consensus members will produce a leader_reselection vote. If there are $\geq 2f + 1$ leader reselection votes, the consensus will return to clock = 0 and reselect a new leader.

(4) *Clock $\geq$ 4λ:* The leader will produce a BLS collective signature for the next round votes, which indicates that at least $\geq 2f + 1$ have been ready for the next round. After receiving a BLS collective signature for the next round votes, the NetChain node will turn into the next round. If there are $\geq 2f + 1$ leader reselection votes, the consensus will return to clock = 0 and reselect a new leader.

V. Design of Bilateral Evaluation Mechanism Based on Game Theory

In multi-domain slice cooperation architecture, multi-domain network service providers may conduct malicious or selfish behaviors due to economic benefits. For example, the network service provider may overstate its available network resources (network, storage, and computing) or provide less network resources for users, resulting in the violation of SLA and the degrading of QoE/QoS. Therefore, there is network resource capacity gap between the provided resource capacity and the user's experienced network resource capacity such as the bandwidth, etc [39]. The network slice user and domain network service provider may not always agree with each other on the provided network resource capacity, which is the key factor to ensure QoE/QoS of multi-domain slice.

To prevent this from happening, NetChain presents a bilateral evaluation mechanism based on game theory in Fig. 3, which introduces credibility and reputation value to quantify the trust of a slice user and network resource provider. The

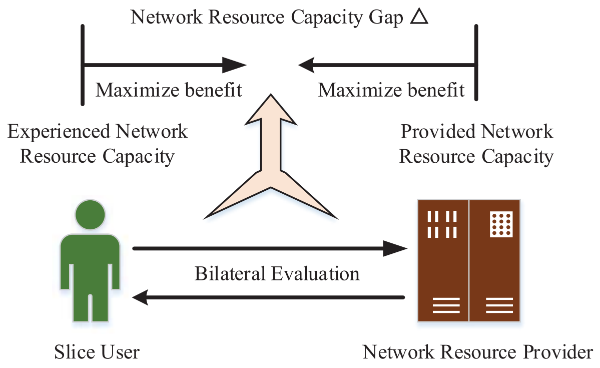
Network Resource Capacity Gap Δ
Maximize benefit
Maximize benefit
Experienced Network Resource Capacity
Provided Network Resource Capacity
Bilateral Evaluation
Slice User
Network Resource Provider

Fig. 3. Bilateral Evaluation Mechanism.

credibility and reputation value is decided by each other's cumulative evaluation, which will directly influence their economic benefits. Therefore, both of them want to maximize their benefits by maximizing their credibility and reputation value. The key insight of bilateral evaluation is to let the network resource gap between slice user and network resource providers reach Nash Equilibrium, eliminating the network resource capacity gap and suppressing the malicious behaviors in the system.

$$P_i^c = \sum_{\kappa=1}^{k} \left(\omega_\kappa * P^c_{i,\kappa}\right) + \omega_{k+1} * P^c_{i,k+1},\ \omega_{\kappa+1} > \omega_\kappa, \quad (9)$$

where

$$P^c_{i,\kappa} = 1/\left(1 + \gamma * e^{-\alpha * P^r_{i,\kappa}}\right), \quad (10)$$

$$k = \left\lfloor \frac{r}{\nu} \right\rfloor. \quad (11)$$

As is shown in (9), P_i^c is the credibility of a network resource provider i, which is the sum of $P^c_{c,\kappa}$ in each time period. $P^c_{c,\kappa}$ is decided by its reputation value $P^c_{r,\kappa}$ defined in (10). In (10), NetChain uses the sigmoid function to map the reputation value $P^r_{i,\kappa}$ into a probability $P^c_{i,\kappa}, (0 < P^c_{i,\kappa} < 1)$, α is the step size. Here, γ is used to ensure that the credibility value of $P^c_{c,\kappa}$ cannot reach to 1 in a short time since the network resource providers may be turn to negative in contributing their network resource once its credibility value of $P^c_{i,\kappa}$ reach to 1. To inspire that the network resource provider will continue to contribute the network resource without conducting malicious behaviors, the time from the system initiation to the current time is equally divided into the same period of ν. The total number of previous time periods is k, which is defined in (11). We take the integer part of the lower boundary of k, ensuring that it is an integer. Credibility weight ω_κ are assigned in chronological order. The closer to the current time period $\kappa + 1$, the greater the credibility weight ω_κ. For example, $\omega_{\kappa+1}$= 2/3; $\omega_\kappa = (2/3)^2$,..., $\omega_1 = (2/3)^k$. This can ensure that NetChain credibility is mainly decided by accumulative bilateral evaluations in several latest time periods, inspiring them to continue to contributing network resource. The time period is quantified by a certain range of rounds ν since NetChain executes in rounds. For example, $\nu = 100,000$, i.e., the period is 100,000 times the consensus latency. The r represents the monotonically increasing round number.

$$S_j^c = \sum_{\kappa=1}^{k} \left(\omega_\kappa * S^c_{j,\kappa}\right) + \omega_{k+1} * S^c_{j,k+1}, \quad (12)$$

where

$$S^c_{j,\kappa} = 1/\left(1 + \gamma * e^{-\alpha * S^r_{j,\kappa}}\right). \quad (13)$$

Similar to the definition of network resource provider, in (12) and (13), S_c^j is the credibility and $S^r_{j,\kappa}$ is the reputation value of a slice user j.

$$P^r_{i,\kappa} = P^{r-1}_{i,\kappa} + S_j^c * e^{-\beta*\left(P^{cp}_{i,\kappa} - S^{cp}_{j,\kappa}\right)},\ P^{cp}_{i,\kappa} \geq S^{cp}_{j,\kappa}, \quad (14)$$

$$S^r_{j,\kappa} = S^{r-1}_{j,\kappa} + P_i^c * e^{-\beta*\left(P^{cp}_{i,\kappa} - S^{cp}_{j,\kappa}\right)},\ P^{cp}_{i,\kappa} \geq S^{cp}_{j,\kappa}, \quad (15)$$

where

$$P^{cp}_{i,\kappa} - S^{cp}_{j,\kappa} = p_c * \left(C^{cp}_{i,\kappa} - C^{cp}_{j,\kappa}\right) + p_s * \left(S^{cp}_{i,\kappa} - S^{cp}_{j,\kappa}\right) + p_l * \left(L^{cp}_{i,\kappa} - L^{cp}_{j,\kappa}\right). \quad (16)$$

The bilateral evaluation mechanism based on game theory is illustrated in (14) and (15). After the multi-domain network slicing service is completed, the slice user and the service provider will evaluate each other. In each bilateral evaluation, reputation value increment of network resource provider $P^r_{i,\kappa}$ is decided by the credibility of slice user and the network resource capacity gap, i.e., the subtraction of network resource capacity provided by provider $P^{cp}_{i,\kappa}$ and network resource capacity measured by slice user $S^{cp}_{j,\kappa}$. β is the step size, influencing the reputation value increment. Vice versa, the reputation value increment of slice user $S^r_{j,\kappa}$ in each bilateral evaluation is decided by the credibility of network provider and the network resource capacity gap. Correspondingly, the network resource capacity gap is explicitly shown in (16), including the computation capacity gap, storage capacity gap, and bandwidth gap quantified by its corresponding price.

$$\Delta = P^{cp}_{i,\kappa} - S^{cp}_{j,\kappa} = arg\ max\ P^r_{i,\kappa}. \quad (17)$$

$$\Delta = P^{cp}_{i,\kappa} - S^{cp}_{j,\kappa} = arg\ max\ S^r_{j,\kappa}. \quad (18)$$

As is shown in (17) and (18), both of slice user and network resource provider want to maximize their benefits by maximizing their credibility and reputation value. The smaller the network resource capacity gap, the closer the reputation value is to 1 in each bilateral evaluation. In a real network environment, a user tends to choose a network resource provider with the higher reputation value to ensure the QoE of network slice, this prompts each network resource provider to get the highest reputation value by minimizing the network resource capacity gap. On the other side, the multi-domain slice user also tends to maximize its influence to the network resource providers and maximize its reputation value by minimizing the network resource capacity gap. Therefore, the best strategy will be $\Delta = P^{cp}_{i,\kappa} - S^{cp}_{j,\kappa} = 0$, reaching Nash Equilibrium if both of them are rational. The bilateral evaluation records will be stored in blockchain, which are transparent and tamper-proofing. The bilateral evaluation mechanism is conducted after the end of the slice life circle, which does not influence the performance of multi-domain orchestration. The mechanism could greatly suppress the malicious behaviors of users and network resource providers during multi-domain network slice orchestration, guaranteeing the QoE/QoS and fairness. At the same time, the bilateral evaluation mechanism will inspire the network resource provider to consistently share their available network resource.

VI. Implementation and Evaluation

The prototype of NetChain is implemented based on the Microsoft CCF platform [18], using Open Enclave SDK. To simulate a realistic and globally distributed deployment, we deploy the NetChain prototype on 4 Microsoft Azure's confidential computing virtual machines distributed in the eastern United States and western Europe. Each VM (DC4s_v2) is confugured with 4 vCPU, 16G memory, and 30G Solid-State Drive. We measure the latency and network bandwidth between VMs using a network performance measurement tool Iperf. The bandwidth between VMs is about 1.8 Gbits/sec within the same region and 0.3 Gbits/sec across different regions on average. The communication latency is about 2ms within the same region and about 86ms across different regions. In the experiments, $\lambda = 86$ ms to guarantee that all the messages can be received within λ, which can be adjusted according to the network slice in a specific environment. The overall number of NetChain nodes simulated by the VMs varies from 4 to 18, evaluating the performance with the changes of network size. Each NetChain node represents a network resource provider in real environment. Besides, we evaluate the security of NetChain and compare it with existing baseline solutions.

A. Performance Evaluation

1) Multi-Domain Slice Orchestration Latency: Multi-domain slice orchestration latency reflects the time required from sending network slicing requests to storing the corresponding results in the blockchain, including the network slice orchestration latency and the consensus latency. The smaller the latency, the more frequently multi-domain slice orchestration transaction can be processed. We evaluate the multi-domain slice orchestration latency performance with the changes of network size, i.e., the node number changes from 4 to 18. The experimental results are shown in Fig. 4, the latency increases with the increment of network size since it needs more time to make consensus and orchestrate the multi-domain network resource. The multi-domain slice orchestration latency is less than 4s when there are 18 NetChain nodes participating in consensus.

To ensure the scalability of NetChain, we present a novel consensus algorithm called CoNet which randomly selects a certain number of nodes to participate in the consensus process, not all of them. Therefore, NetChain can scale up to hundreds of nodes without trading off too much performance. Currently, the major telecom operating companies located in different countries in total are less than one hundred. As a result, NetChain can aggregate the available network resources of operators around the world, providing agile end-to-end multi-domain slice orchestration service globally.

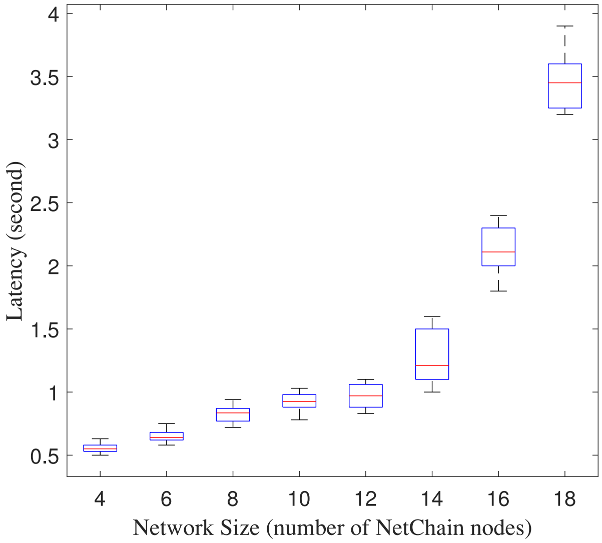

Fig. 4. Multi-domain network slice orchestration latency in NetChain.

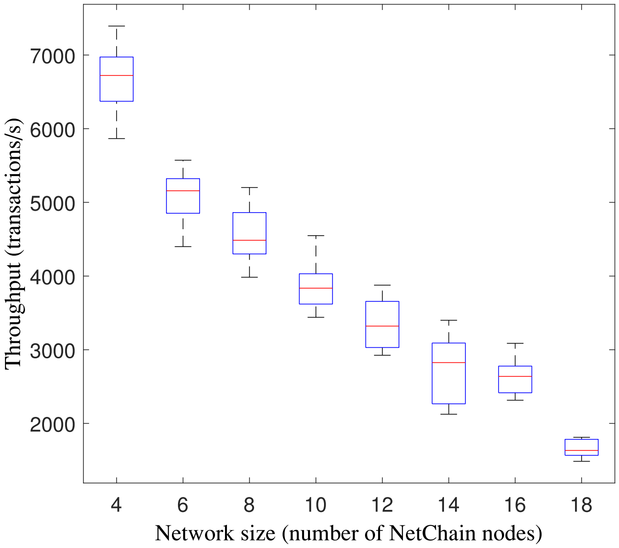

Fig. 5. Multi-domain network slice orchestration throughput in NetChain.

2) Multi-Domain Slice Orchestration Throughput: Throughput reflects the maximum rate of multi-domain slice orchestration requests that can be processed in NetChain. The higher the throughput, the more requests can be processed per second. The same with latency, we evaluate the multi-domain slice orchestration latency performance with the changes of network size. The experimental results are illustrated in Fig. 5, the throughput decreases as the increment of network size since it needs more time to make consensus and orchestrate multi-domain network resources.The multi-domain slice orchestration throughput is more than 1,500 when there are 18 NetChain nodes participating in the consensus. Network slicing is a logical and customized network running on the same shared physical infrastructure. The network slice request per second is sparser in a real network environment, especially for the multi-domain network slice. Therefore, the throughput performance of NetChain is good enough to meet multi-domain network slice requirements.

B. Security Evaluation

1) Consensus Security Analysis: In a blockchain-based system, consensus security is critical for system security. To ensure the scalability of NetChain, we presents a novel consensus algorithm which randomly selects a certain number of nodes to participate in consensus process, not all of them. In asynchronous consensus protocols, $3f + 1$ is the minimum number of replicas that allow an asynchronous system to provide the safety and liveness properties when up to f replicas are faulty. We assume that the proportion of malicious nodes is p, and the number of malicious nodes is n*p. Therefore, the system has no chance to be controlled by the adversary if $n * p \leq 2N_c/3$. However, the consensus process may fail when there are more than one-third N_c malicious nodes in the system. We need to fully analyze the relationship between the number of consensus members N_c and overall number n in influencing the probability of consensus failure or controlled by the adversary. The consensus failure probability C_f is illustrated in (19). Correspondingly, the system can be controlled by the adversary if $2N_c/3 \leq n * p$, and the probability that the system is controlled by the adversary is shown in (20).

$$C_f = \begin{cases} 0, \quad n * p < N_c/3 \\ \sum_{i=N_c/3}^{n*p} \frac{C_{n*p}^{i} * C_{n-n*p}^{N_c-i}}{C_n^{N_c}}, \quad N_c/3 \leq n * p \leq 2N_c/3 \\ \sum_{i=N_c/3}^{2N_c/3} \frac{C_{n*p}^{i} * C_{n-n*p}^{N_c-i}}{C_n^{N_c}}, \quad 2N_c/3 \leq n * p \end{cases} \tag{19}$$

$$C_c = \begin{cases} 0, \quad n * p \leq 2N_c/3 \\ \sum_{i=2N_c/3}^{n*p} \frac{C_{n*p}^{i} * C_{n-n*p}^{N_c-i}}{C_n^{N_c}}, \quad 2N_c/3 \leq n * p \leq N_c \\ \sum_{i=2N_c/3}^{N_c} \frac{C_{n*p}^{i} * C_{n-n*p}^{N_c-i}}{C_n^{N_c}}, \quad N_c \leq n * p \end{cases} \tag{20}$$

We conduct simulation to analyze consensus failure probability with the changes in network size. The parameter settings are as follows, $N_c = 2n/3, N_c/3 \leq n * p \leq 2N_c/3, n = 18, \ldots, 288$ to simulate different network size. The simulation result is shown is Fig. 6 , and we can find that the consensus failure probability decreases with the number of network size increases. The more nodes in total, the lower the probability of consensus failure probability and the safer the system, when the proportion of malicious nodes is fixed. To guarantee that there is no probability that the system can be controlled by the malicious attacker, we need to ensure $n * p < 2N_c/3$. In addition, the N_c should be adjusted according to malicious node number n*p in a real environment to make a balance between system performance and security.

2) The Effectiveness of Bilateral Evaluation Based on Game Theory in Ensuring QoE/QoS and Fairness: In the existing solutions, there is no effective mechanism to ensure QoE/QoS and fairness during multi-domain network slice orchestration. Our proposed NetChain presents a bilateral evaluation mechanism based on game theory to ensure QoE/QoS and fairness by suppressing the malicious behaviors during multi-domain network slice orchestration. The network resource capacity gap is a critical parameter in reflecting whether or not a network resource provider conducts malicious behaviors such as overstate its available network resource. The network resources capacity gap directly affects the credibility of the network resource providers, thereby affecting their economic interests. The credibility of the slice user is similar to the network resource provider. Therefore, both of them want to maximize their benefits by maximizing their credibility and reputation value. The key insight of bilateral evaluation is to let the network resource gap between slice user and network resource providers reach Nash Equilibrium 0.

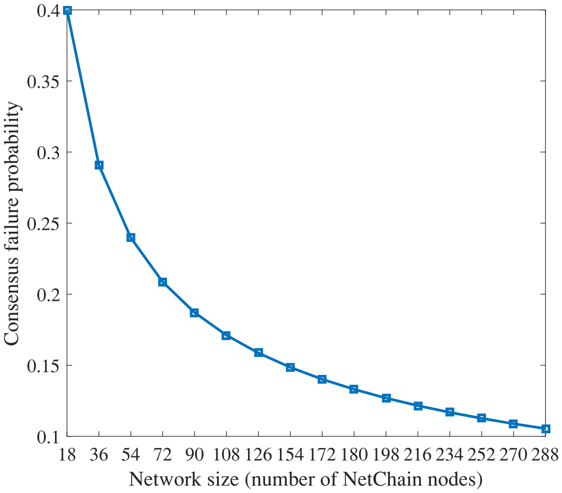

Fig. 6. Consensus failure probability changes with network size when the proportion of malicious nodes is fixed.

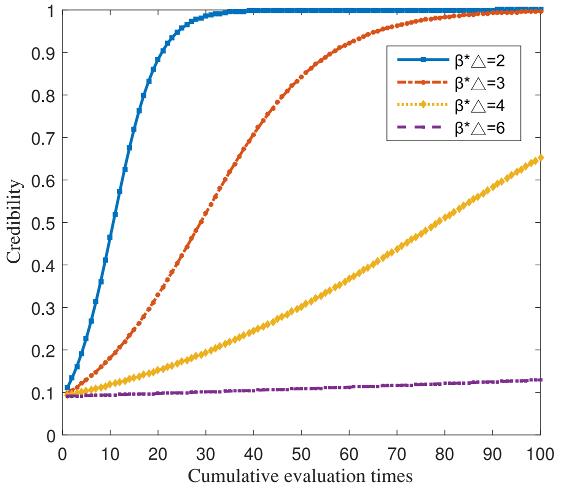

Fig. 7. Effectiveness validation of bilateral evaluation mechanism.

To validate the effectiveness of the proposed bilateral evaluation mechanism based on game theory in suppressing malicious behaviors and eliminating the network resource capacity gap, we conduct the following simulation. The parameter settings are as follows, $\gamma = 10$, $\alpha = 2$, $P_{i,\kappa}^{r} = 0$, $S_j^c = 0.8$. we adjust $\beta * \Delta = 2, 3, 4, 6$ to simulate the different network resource capacity gap scenarios. As is illustrated in Fig. 7, the credibility of network resource provider increases very slowly with the accumulation of bilateral evaluation times if

TABLE I
COMPARISON OF NETCHAIN AND EXISTING BASELINE SOLUTIONS

Contents	NetChain	[23]	[18]
Latency(s)	<4s	<7s	$\approx$ 0.5s
Throughput (transactions/s)	>1500	15–45	$\approx$ 100
Security	high	medium	medium
Scalability	high	high	low
Privacy-preserving	yes	no	no

the network resource gap is big such as $\beta * \Delta = 6$. Otherwise, i.e., $\beta * \Delta = 2$, the credibility will increase exponentially as the accumulation of bilateral evaluation times. The parameter credibility directly influences the economic benefit in a real environment. In theory, the network resource gap will be 0, reaching Nash Equilibrium based on game theory. Therefore, the bilateral evaluation mechanism based on game theory is effective in suppressing malicious behaviors.

C. Comparison of NetChain With Existing Baseline Solutions

As is shown in Table I, we compare the NetChain with existing baseline blockchain-based multi-domain network slice orchestration solutions in terms of latency, throughput, security, scalability, and privacy-preserving. The solution [17] and [18] are built on existing blockchain platforms Ethereum and Hyperledger Fabric respectively.

Latency and Throughput: Latency and throughput performance reflect the processing ability of multi-domain network resource slice orchestration transaction. The performance of [17] and [18] is evaluated when there are 5 nodes or organizations participated in consensus. The latency of NetChain is less than 4s, which is about 3.5s and 0.5s on average when there are 18 and 5 nodes participating in the consensus. Therefore, the latency of NetChain is less than [17] with 7s and similar to [18] with 0.5s. However, the [18] is built on Raft-based Hyperledger Fabric, in which the latency and throughput performance will decrease sharply with the increase of network size. The throughput of NetChain is larger than 1,500, which is better than [17] with dozens of transactions per second and [18] with 100 transactions per second respectively. In conclusion, NetChain performs well both in latency and throughput.

Security: Consensus security is critical for a blockchain-based system. The solution in [18] is based on Raft consensus in which the leader selection process is fixed, making it vulnerable to DoS attack. Besides, Raft cannot tolerate malicious behaviors. The solution in [17] relies on Ethereum, which is based on PoW and Ethereum 2.0 will move to PoS. Although PoS and PoW both work in a fintech setup and have high security, they still face forking risks that may result in data inconsistency. The forking probability is based on the proportion of malicious computing power or malicious stake. The forking probability of PoW in bitcoin is $P = 0.0002428$ [40], when the proportion of malicious computing power is 0.1. The PoS is similar to PoW, and the only difference is that PoS is based on stake (money). Forking is very harmful to the fairness and security of multi-domain network slice orchestration. The consensus of NetChain is derived from PBFT, which is non-forking since the consensus is based on the voting of majority nodes.

PoW and PoS can mitigate Sybil attacks based on computing power or money (stake), respectively. It is extremely hard for the attacker to gather a large portion of tokens or computing power to launch Sybil attacks. However, PoW and PoS run in an open system, an attacker can still maliciously create a large number of Sybil nodes with new node IDs to disrupt the network, threatening system security. NetChain presents an authentication and access control mechanism to prevent Sybil attacks. In NetChain, the consensus member and a leader are randomly selected and changed every round to enhance security. Moreover, NetChain proposes a bilateral evaluation mechanism based on game theory to ensure QoE/QoS and fairness by suppressing malicious behaviors in the system.

Scalability: The solution [17] is built on Ethereum, which have good scalability. The solution in [18] is based on Raft consensus, which has poor scalability. Raft needs all the nodes in the system to participate in consensus, in which the communication overhead increase and the performance decrease sharply as the increment of the number of nodes. Different from Raft, NetChain randomly selects a certain number of nodes as consensus members in each round to participate in the consensus, not all of them. Besides, NetChain introduces BLS signature during the consensus process to reduce communication overhead and improve the performance. Therefore, NetChain performs well in scalability.

Privacy-Preserving: Both of solutions in [17] and [18] need all the network resource providers to transparently share their private network resource information, lacking privacy consideration. NetChain uses TEE to provide a privacy-preserving environment for multi-domain network slice orchestration information, eliminating the privacy concern of network resource providers to share their available network resource and making NetChain easily to be deployed in a real environment.

VII. DISCUSSION

This section provides a full discussion about the advantages of NetChain compared with the existing solutions. Moreover, we analyze the limitations of SGX-based privacy-preserving solution and the potential alternative in the future. Finally, we discuss the compatibility design of NetChain.

A. NetChain's Advantages Compared With Existing Solutions

Privacy-Preserving With High Performance: The current privacy-preserving multi-domain slice orchestration solutions are either provide weak privacy protection or based on time consuming MPC [41] which is only applicable to limited scenarios. NetChain is based on TEE to provide privacy-preserving and agile multi-domain slice orchestration, which has fundamental advantages compared with the current solutions. TEE can provide a fully isolated environment called an enclave that prevents other software applications, the operating system, and the host owner from tampering with or learning the state of an application running in the TEE. The remote authentication in TEE can ensure that the data passed into the SGX through the encrypted channel will be processed according to the network slice consumers' and operators' expected policy, ensuring that the information is not leaked. TEE is based on hardware and can guarantee the information privacy in NetChain with high performance.

High Scalability and Security Without Relying on Any Third-Party: The current blockchain-based multi-domain network slice solutions simply rely on existing third-party platforms such as Ethereum or Hyperledger Fabric, which face faces security risks or scalability challenges. For example, Ethereum adopts PoW as the consensus algorithm and is moving to PoS in the future. Although PoS and PoW both work in a fintech setup and have high security, they still face forking risks. Indeed, PoW and PoS can mitigate Sybil attacks based on computing power and money (stake), respectively. It is extremely hard for the attacker to gather a large portion of tokens in the system to launch Sybil attacks. However, Ethereum runs in an open system, an attacker can still maliciously create a large number of Sybil nodes with new node IDs to disrupt the network, threatening the system security. Besides, the PoS needs all the participants to deposit a lot of stakes (money) and PoW will consume a lot of electricity and computing power. Hence, a blockchain-based multi-domain network resource sharing and slice orchestration architecture based on Ethereum is costly, making it difficult to deploy in a real environment. Finally, if a multi-domain network slice orchestration architecture is built on a third-party blockchain platforms such as Ethereum, the transaction processing performance will be greatly influenced by their network situation, resulting in poor QoS/QoE. For example, in 2017, the popularity of the CryptoKitties game in Ethereum causes the network to become heavily congested, slowing transaction processing significantly.

Hyperledger Fabric is a permissioned blockchain system, which is commonly based on Raft or PBFT. However, both of them has poor scalability and the performance decreases exponentially as the number of consensus nodes increases. The typical application scenario of them is limited to only a few or no more than dozens of nodes. Indeed, we can only choose a small consortium of nodes from a large number of distinct operators to participate in governance and consensus in multi-domain network slice orchestration architecture. However, if only a small number of operators participated in the consensus, they may conduct malicious behaviors, affecting the fairness and credibility of the system. Fairness and credibility are crucially important for a multi-domain network slice orchestration architecture. Therefore, we insist that all the operators should have a chance to participate in the governance and consensus. Besides, the leader selection process in PBFT is fixed and can be predicted, making it vulnerable to DoS attacks.

In conclusion, to ensure stable transaction processing performance and reduce the security attack surface, we insist that a multi-domain network slice orchestration architecture should not rely on any third-party blockchain platform. To cope with challenges and risks in the current blockchain-based solutions, NetChain presents a novel consensus algorithm CoNet, which randomly selects a certain number of nodes to participate in consensus to guarantee scalability and security.

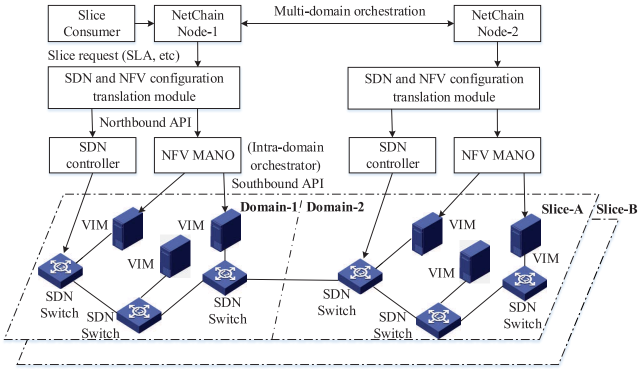

Fig. 8. Compatibility design of NetChain.

Moreover, a bilateral evaluation mechanism based on game theory is proposed to guarantee fairness and QoE by suppressing malicious behaviors in the system. Therefore, the proposed NetChain performs better in terms of security and scalability, compared with the current blockchain-based multi-domain network slice orchestration solutions.

B. Limitations of SGX and Future Alternatives

Limitations of SGX-Based NetChain: NetChain is based on the assumption that SGX must be trusted, which provides a fully isolated environment that prevents other software applications, the operating system, and the host owner from tampering with or even learning the state of an application running in the TEE. However, most of the Intel SGX implementation is not transparent to the customer. Therefore, it is hard for the customer to verify the correctness of the implementation and fully trust Intel SGX. Besides, several other challenges remain unaddressed such as side-channel attacks and secure I/O.

Future Privacy-Preserving Alternatives: In the future, the open-source TEE project such as Keystone [42] may be a potential alternative solution to cope with the challenges and security risks in Intel SGX. Keystone is based on open-source hardware ISA RISC-V [43], in which the implementation can be verified and trusted. Besides, the combination of TEE and MPC can provide a stronger and more flexible privacy-persevering solution since they have complementary properties. TEE can be used to ensure the security of MPC, while MPC can be adopted to enhance the privacy of TEE.

C. Compatibility Consideration

Compatibility is critical for the widespread deployment of NetChain. Therefore, NetChain is designed as an overlay architecture and can be incrementally deployed, which needs no changes in the northboud and southbound interface of SDN controller and NFV MANO.

The compatibility design is illustrated in Fig. 8. There is a NetChain node added in each network administrative domain, which is in charge of a multi-domain network slice creation, update, and revocation. The NetChain node orchestrates an end-to-end multi-domain slice based on the available multi-domain network status information when it receives the request from a slice consumer. Moreover, an SDN and

NFV configuration translation module is designed, translating the end-to-end network slice orchestration output information into executable SDN and NFV configuration files. Finally, the multi-domain network slice will be created based on the SDN and NFV configuration files by the corresponding SDN controller and NFV MANO. For example, in ONOS (Open Network Operating System) [44], the multi-domain slice orchestration output will be the input of the ONOS application to configure SDN switch through OpenFlow in the network layer. For the NFV controller such as Tacker, the multi-domain slice orchestration output will be sent to Tacker through REST API. Then, the Generic VNF Manager and an NFV Orchestrator will deploy operate Network Services and Virtual Network Functions on the NFV infrastructure.

VIII. Conclusion

In this paper, we present a privacy-preserving multi-domain slice orchestration architecture called NetChain based on blockchain and TEE, eliminating the security risks and challenges such as privacy disclosure, single-point failure, and no effective mechanism to ensure QoE/QoS in existing solutions. A novel consensus algorithm is designed to enhance the security and scalability of the architecture. Moreover, a bilateral evaluation mechanism based on game theory is proposed to ensure QoE/QoS and fairness by suppressing malicious behaviors during multi-domain slice orchestration. Finally, the NetChain is implemented and evaluated on the Microsoft Azure Cloud with confidential computing. The experimental results show that NetChain has good performance and outperforms the current blockchain-based solutions in terms of privacy-preserving and security.

For future work, we plan to combine the advantages of TEE and MPC to provide stronger privacy-persevering and decentralized trustworthy multi-domain network slice orchestration.

References

[1] P. Rost *et al.*, “Mobile network architecture evolution toward 5G,” *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 84–91, May 2016.

[2] F. Poltronieri, L. Campioni, R. Lenzi, A. Morelli, N. Suri, and M. Tortonesi, “Secure multi-domain information sharing in tactical networks,” in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Los Angeles, CA, USA, 2018, pp. 1–6.

[3] W. Zhang, D. Yang, Y. Xu, X. Huang, J. Zhang, and M. Gidlund, “DeepHealth: A self-attention based method for instant intelligent predictive maintenance in industrial Internet of Things,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5461–5473, Aug. 2021.

[4] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, “Network function virtualization: State-of-the-art and research challenges,” *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

[5] H. Moens and F. De Turck, “Customizable function chains: Managing service chain variability in hybrid NFV networks,” *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 4, pp. 711–724, Dec. 2016.

[6] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, “Network slicing in 5G: Survey and challenges,” *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, May 2017.

[7] F. Song, L. Li, I. You, and H. Zhang, “Enabling heterogeneous deterministic networks with smart collaborative theory,” *IEEE Netw.*, vol. 35, no. 3, pp. 64–71, May/Jun. 2021.

[8] Y. Qu, S. Yu, W. Zhou, S. Peng, G. Wang, and K. Xiao, “Privacy of things: Emerging challenges and opportunities in wireless Internet of Things,” *IEEE Wireless Commun.*, vol. 25, no. 6, pp. 91–97, Dec. 2018.

[9] Y. Qu, S. Yu, L. Gao, W. Zhou, and S. Peng, “A hybrid privacy protection scheme in cyber-physical social networks,” *IEEE Trans. Comput. Soc. Syst.*, vol. 5, no. 3, pp. 773–784, Sep. 2018.

[10] N. F. S. De Sousa, D. A. L. Perez, R. V. Rosa, M. A. S. Santos, and C. E. Rothenberg, “Network service orchestration: A survey,” *Comput. Commun.*, vols. 142–143, pp. 69–94, Jun. 2019.

[11] T. Taleb, I. Afolabi, K. Samdanis, and F. Z. Yousaf, “On multi-domain network slicing orchestration architecture and federated resource control,” *IEEE Netw.*, vol. 33, no. 5, pp. 242–252, Sep./Oct. 2019.

[12] D. Dietrich, A. Abujoda, A. Rizk, and P. Papadimitriou, “Multi-provider service chain embedding with Nestor,” *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 1, pp. 91–105, Mar. 2017.

[13] R. A. Addad, M. Bagaa, T. Taleb, D. L. C. Dutra, and H. Flinck, “Optimization model for cross-domain network slices in 5G networks,” *IEEE Trans. Mobile Comput.*, vol. 19, no. 5, pp. 1156–1169, May 2020.

[14] T. Mano, T. Inoue, D. Ikarashi, K. Hamada, K. Mizutani, and O. Akashi, “Efficient virtual network optimization across multiple domains without revealing private information,” *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 477–488, Sep. 2016.

[15] A. Francescon, G. Baggio, R. Fedrizzi, R. Ferrusy, I. G. Ben Yahiaz, and R. Riggio, “X–MANO: Cross–domain management and orchestration of network services,” in *Proc. IEEE Conf. Netw. Softw. (NetSoft)*, Bologna, Italy, 2017, pp. 1–5.

[16] K. D. Joshi and K. Kataoka, “pSMART: A lightweight, privacy-aware service function chain orchestration in multi-domain NFV/SDN,” *Comput. Netw.*, vol. 178, Sep. 2020, Art. no. 107295.

[17] R. V. Rosa and C. E. Rothenberg, “Blockchain-based decentralized applications for multiple administrative domain networking,” *IEEE Commun. Stand. Mag.*, vol. 2, no. 3, pp. 29–37, Sep. 2018.

[18] N. Afraz and M. Ruffini, “5G network slice brokering: A distributed blockchain-based market,” in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Dubrovnik, Croatia, Apr. 2020, pp. 23–27.

[19] G. Wood *et al.*, “Ethereum: A secure decentralised generalised transaction ledger,” Ethereum Project, Zug, Switzerland, Yellow Paper, vol. 151, pp. 1–32, 2014.

[20] E. Androulaki *et al.*, “Hyperledger fabric: A distributed operating system for permissioned blockchains,” in *Proc. 13th EuroSys Conf.*, 2018, p. 30.

[21] R. Cheng *et al.*, “Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts,” in *Proc. IEEE Eur. Symp. Security Privacy (EuroS&P)*, Stockholm, Sweden, 2019, pp. 185–200.

[22] F. Song, Y.-T. Zhou, Y. Wang, T.-M. Zhao, I. You, and H.-K. Zhang, “Smart collaborative distribution for privacy enhancement in moving target defense,” *Inf. Sci.*, vol. 479, pp. 593–606, Apr. 2019.

[23] D. Ongaro and J. Ousterhout, “The Raft consensus algorithm,” Lecture Notes CS 190, Stanford Univ., Stanford, CA, USA, 2015.

[24] S. Johnson, V. Scarlata, C. Rozas, E. Brickell, and F. Mckeen, “Intel–software guard extensions: Epid provisioning and attestation services,” Intel, Santa Clara, CA, USA, White Paper, vol. 1, p. 119, 2016.

[25] A. Kate and I. Goldberg, “Distributed key generation for the Internet,” in *Proc. 29th IEEE Int. Conf. Distrib. Comput. Syst.*, Montreal, QC, Canada, 2009, pp. 119–128.

[26] D. Schultz, B. Liskov, and M. Liskov, “MPSS: Mobile proactive secret sharing,” *ACM Trans. Inf. Syst. Security*, vol. 13, no. 4, pp. 1–32, 2010.

[27] S. Bano *et al.*, “SoK: Consensus in the age of blockchains,” in *Proc. 1st ACM Conf. Adv. Financ. Technol.*, 2019, pp. 183–198.

[28] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, “The Honey Badger of BFT protocols,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2016, pp. 31–42.

[29] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, “A secure sharding protocol for open blockchains,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2016, pp. 17–30.

[30] E. Kokoris-Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, “Enhancing bitcoin security and performance with strong consistency via collective signing,” in *Proc. 25th USENIX Security Symp. (USENIX Security)*, 2016, pp. 279–296.

[31] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, “OmniLedger: A secure, scale-out, decentralized ledger via sharding,” in *Proc. IEEE Symp. Security Privacy (SP)*, San Francisco, CA, USA, 2018, pp. 583–598.

[32] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, “Algorand: Scaling byzantine agreements for cryptocurrencies,” in *Proc. 26th Symp. Oper. Syst. Principles*, 2017, pp. 51–68.

[33] B. Alangot, M. Suresh, A. S. Raj, R. K. Pathinarupothi, and K. Achuthan, “Reliable collective cosigning to scale blockchain with strong consistency,” in *Proc. Netw. Distrib. Syst. Security Symp. (DISS) NDSS*, 2018, pp. 1–6.

[34] G. He, W. Su, S. Gao, and J. Yue, "TD-Root: A trustworthy decentralized DNS root management architecture based on permissioned blockchain," *Future Gener. Comput. Syst.*, vol. 102, pp. 912–924, Jan. 2020.

[35] G. He, W. Su, S. Gao, J. Yue, and S. K. Das, "ROAchain: Securing route origin authorization with blockchain for inter-domain routing," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 2, pp. 1690–1705, Jun. 2021.

[36] E. Syta *et al.*, "Keeping authorities 'honest or bust' with decentralized witness cosigning," in *Proc. IEEE Symp. Security Privacy (SP)*, San Jose, CA, USA, 2016, pp. 526–545.

[37] G. Bracha and S. Toueg, "Asynchronous consensus and broadcast protocols," *J. ACM*, vol. 32, no. 4, pp. 824–840, 1985.

[38] E. Syta *et al.*, "Scalable bias-resistant distributed randomness," in *Proc. IEEE Symp. Security Privacy (SP)*, San Jose, CA, USA, 2017, pp. 444–460.

[39] Y. Li, K.-H. Kim, C. Vlachou, and J. Xie, "Bridging the data charging gap in the cellular edge," in *Proc. ACM Spec. Interest Group Data Commun.*, 2019, pp. 15–28.

[40] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[41] R. Cramer, I. B. Damgård, and J. B. Nielsen, *Secure Multiparty Computation*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[42] D. Lee, D. Kohlbrenner, S. Shinde, D. Song, and K. Asanović, "Keystone: An open framework for architecting TEEs," 2019. [Online]. Available: arXiv:1907.10119.

[43] A. S. Waterman, "Design of the RISC-V instruction set architecture," Ph.D. dissertation, Dept. Comput. Sci., Univ. California, Berkeley, CA, USA, 2016.

[44] P. Berde *et al.*, "ONOS: Towards an open, distributed SDN OS," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw.*, 2014, pp. 1–6.

Guobiao He (Student Member, IEEE) received the M.S. degree in communication engineering from Beijing Jiaotong University, Beijing, China, in 2014, where he is currently pursuing the Ph.D. degree in communications and information system. His specific areas of research interest mainly focus on blockchain, cyber security, and future Internet architecture.

Wei Su received the Ph.D. degree in communication and information systems from Beijing Jiaotong University in 2008, where he is currently a Professor. He is mainly engaged in researching key theories for the next generation Internet and has taken part in many national projects, such as the National Basic Research Program (also called the 973 Program) and the National Natural Science Foundation of China.

Shuai Gao (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in communication and information systems from Beijing Jiaotong University in 2001, 2004, and 2010, respectively. From 2015 to 2016, he was a Visiting Scholar with the University of Arizona. He is currently a Full Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University. He is also with the PCL Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen, China. His research interests are in the areas of Internet architecture, sensor networks, and mobile Internet.

Ningchun Liu (Student Member, IEEE) is currently pursing the Ph.D. degree in communication and information systems with National Engineering Lab for Next Generation Internet Technologies, Beijing Jiaotong University, China. His current research interests include information centric networking, software defined networking, and applied cryptography.

Sajal K. Das (Fellow, IEEE) is a Professor of Computer Science and the Daniel St. Clair Endowed Chair with Missouri University of Science and Technology, USA. He holds five U.S. patents and has coauthored four books. He has over 35,000 Google Scholar citations with H-Index of 93. His research interests include wireless sensor networks, mobile and pervasive computing, crowdsensing, cyber-physical systems and IoT, smart environments (e.g., smart city, smart grid, and smart health care), cloud computing, cyber security, biological and social networks, and applied graph theory and game theory. He has published extensively in these areas with over 700 research articles in high quality journals and refereed conference proceedings. He is a recipient of ten best paper awards at prestigious conferences, including ACM MobiCom and IEEE PerCom, and numerous awards for teaching, mentoring, and research, including the IEEE Computer Society's Technical Achievement Award for pioneering contributions to sensor networks and mobile computing. He serves as the Founding Editor-in-Chief of *Pervasive and Mobile Computing* (Elsevier), and as an Associate Editor of several journals, including the IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, and *ACM Transactions on Sensor Networks*.