




Blockchain-Based Cache Poisoning Security Protection and Privacy-Aware Access Control in NDN Vehicular Edge Computing Networks

Kai Lei · Junjie Fang · Qichao Zhang · Junjun Lou ·
Maoyu Du · Jiyue Huang · Jianping Wang · Kuai Xu 

Received: 29 August 2019 / Accepted: 2 August 2020 / Published online: 25 August 2020
© Springer Nature B.V. 2020

Abstract Recent advances in artificial intelligence, big data, mobile edge computing and embedded systems have successfully driven the emergence and adoption of smart vehicles and vehicle edge computing which will improve road safety, traffic

congestions, and vehicle exhaust emissions. The high-mobility, ad-hoc network topology, and diverse vehicle-to-everything (V2X) have brought substantial challenges in the TCP/IP-based vehicular networking. Given the unique characteristics and strengths in resilient communication in mobile ad hoc networking environments, named data networking (NDN) has become a natural fit for supporting vehicular edge computing (VEC) as the underlying network architecture. However, a variety of security and privacy challenges remain for developing NDN-based VEC networks such as key management, cache poisoning, access control. In this paper, we introduce a novel blockchain-based security architecture in NDN-based VEC networks to systematically tackle these security challenges. More specifically, we design and implement an efficient blockchain system on NDN by adopting lightweight yet robust delegate consensus algorithm, and carry out extensive experiments to evaluate performance efficiency on key management protocols, cache poisoning defense schemes, and access control strategies for NDN-based VEC networks. To the best of our knowledge, this paper is the first effort to systematically devise practical and efficient blockchain-based security architecture to provide key management, cache poisoning security protection, and privacy-aware access control in NDN VEC networks.

K. Lei · J. Fang · Q. Zhang · J. Lou · M. Du · J. Huang
Shenzhen Key Lab for Information Centric Networking & Blockchain Technology (ICNLAB), School of Electronics and Computer Engineering (SECE), Peking University, 18 Tat Hong Avenue Shek Kip Mei 518055, China

K. Lei
e-mail: leik@pkusz.edu.cn

L. Kai · K. Lei
PCL Research Center of Networks and Communications,
Peng Cheng Laboratory, Shenzhen, China

J. Wang
Multimedia software Engineering Research Centre, City University of Hong Kong, 18 Tat Hong Avenue Shek Kip Mei Kowloon, Hong Kong

J. Wang
e-mail: merc@cityu.edu.hk

K. Xu (✉)
School of Mathematical and Natural Sciences, New College of Interdisciplinary Arts and Sciences, Arizona State University, 4701 W Thunderbird Rd Glendale AZ 85306, USA

K. Xu
e-mail: Kuai.Xu@asu.edu

Keywords NDN VEC network · Blockchain

1 Introduction

The recent rapid development and deployment of self-driving vehicles and vehicle networking for improved driving safety, energy-saving, and traffic management has benefited from the latest advances in artificial intelligence, mobile edge computing and embedded systems. Given the high-mobility and content-oriented characteristics in V2X data communications such as vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and infrastructure-to-vehicle (I2V) communications, NDN has been considered as one of the promising network architectures for VEC networks due to its efficiency on content distributions what is the content rather than hosts, locations and where of data exchanges between content consumers and content publishers [12, 31].

In VEC networks, the majority of nodes, i.e., moving vehicles, have high mobility, which brings substantial challenges for traditional TCP/IP-based Internet in delivering data between mobile vehicles efficiently and maintaining routing consistency in constant topology changes. On the other hand, NDN, an emerging network architecture driven by data consumers, leverages location-independent data names to identify content and effectively acquire and disseminate content by their names. In addition, the caching and data forwarding features of NDN make data transmission more efficient in VEC networking. However, the inherent data security and privacy threats of such content-centric networking solutions remain in the integrations of NDN and VEC networks, thus, it is very critical to develop novel effective systems to enhance the security and privacy of NDN-based VEC networks.

In NDN VEC networks, the communication channel does not exist between fixed two end-points, such as IP-based hosts. Thus, the research focus of enhancing security and privacy is on data packets. NDN requires each named *data* object to be digitally signed by its producer. However, in practice, this security feature poses a challenge for verifying fake content. For example, the root key may be attacked and tampered. When the root key as a trust anchor fails, each site essentially becomes a relatively independent trust domain. With the absence of trust anchors, it is a challenging task to verify the authenticity of the key issued by a large number of independent trust domains. Moreover,

NDN provides content-based security by verifying the signature of each data packet on intermediate nodes, i.e., vehicles. Due to significant computation and communication overheads, vehicles are often unable to carry out such tasks of performing per-packet signature verifications. As a result, content poisoning attacks can be easily carried out, in which attackers inject a large number of tampered and forged packages into the cache of network nodes to isolate users' access to the authentic content. Such attacks can be fatal in NDN VEC networks, e.g., attackers hiding congestion information in a location in tampered or forged interest packages.

In addition to the above security threats on data authenticity and integrity, the challenge of ensuring data confidentiality is also very critical. For example, unauthorized data access on the locations of certain vehicles at a given time is a huge privacy concern. As NDN makes extensive use of the in-network cache, the content produced by the publisher is cached on all the intermediate routers in the network forwarding path. The in-network cache strategy decouples the content from its publisher, and the consumers often obtain a cached copy of the content from intermediate routers. The existing NDN architecture implements access control via a *content-based* encryption scheme, in which a central node such as a server acts as an engine for processing access control logic. However, the centralized access logic execution methods face an inherent trust problem of the authorization engine. Therefore, NDN-based VEC networks require a *trust-based* access control scheme to control unauthorized access to copies of the content that are cached on intermediate nodes.

In this paper, we introduce a blockchain-based security architecture for improving the security and privacy of NDN-based VEC networks. In contrast to traditional centralized system architectures with the risk of the single point of failures and the problem of trust issues in multi-trust domains, the blockchain build trust among multiple untrusted nodes without a centralized third party by making all participating nodes in the network share and verify the data under certain consensus. The open and trustworthy security infrastructure ensures the secure transmission of data via consensus algorithms, and also provides necessary services for different security applications such as designing new mechanisms for key management, content poisoning attack detection, and access control for

address security and privacy challenges in NDN-based VEC networks.

The major contributions of this paper can be summarized as follows:

- We propose a novel blockchain-based security architecture for NDN VEC networks and develop three key mechanisms for key management, cache poisoning detection, and access control for protecting VEC networks for a wide spectrum of security threats.
- We design and implement an efficient blockchain system on NDN by adopting delegate consensus algorithms. More importantly, we systematically evaluate the performance effectiveness, cost and efficiency of the proposed blockchain system on NDN via extensive experimental evaluations on the ndnSIM platform.
- We verified that the proposed key management scheme developed in the proposed blockchain-based security architecture for NDN VEC networks is able to solve the trust security problem caused by single trust anchors.
- Our experimental results show that the cache poisoning detection scheme designed in this paper protects cache poisoning attacks with lightweight computational computation and validation overheads on intermediate nodes.
- The proposed access control scheme supports resource owners to have fine-grained access control over their own resources, and supports the applicants to prove their access rights via the blockchain system.

The remainder of this paper is structured as follows. Section 2 gives a brief introduction to the basic concepts of VEC, NDN and blockchain, and explains why VEC is worthy of attention and why NDN is more suitable as the infrastructure of VEC. Section 3 introduces the system model and key designs components of the proposed blockchain-based architecture for securing NDN VEC networks. Section 4 sheds light on the critical applications of the proposed architecture in key management, content poisoning attack detection and access control, and Section 5 systematically presents the performance and cost of the proposed architecture via extensive experimental evaluations. Section 6 discusses related work, while Section 7 concludes this paper and outlines future work.

2 Background

2.1 Vehicular Edge Computing Networking

The emergence of intelligent vehicle applications has brought many benefits in road traffic safety, driving efficiency, and in-vehicle entertainment. However, these applications have also raised challenges on systematic performance and security for resource-constrained vehicles. To address these challenges, an early study [9] has proposed the initial vehicle cloud computing (VCC) architecture model. Recently, the vehicle networking community has introduced a new model of vehicle edge computing (VEC) [35] for accommodating broad data communications of V2X networking including V2V, V2I, and infrastructure-to-vehicle (I2V).

Compared with cloud-based computing service models, a major benefit of VEC lies in the faster interactive responses provided by the *local* service providers that are physically closer to vehicle nodes, thus guaranteeing the user experience of delay-sensitive applications and reducing the end-to-end latency and transmission cost [7]. In addition, the ubiquitous edge resources in VEC could directly offer vehicles and users many services in driving safety and efficiency such as real-time navigation and local service discovery.

In a typical VEC environment, a variety of nodes could act as service providers including base stations, Internet of Things (IoT) gateways, and vehicles themselves. For example, the base station collects information of surrounding vehicles and performs streaming data analysis to provide accurate and real-time traffic information to vehicles. Similarly, vehicles, experiencing road traffic congestion, could broadcast congestion locations and durations via V2V networks in which nearby vehicles serve as forwarding nodes to disseminate data packets carrying congestion information. On the other hand, selfish or malicious nodes could potentially compromise the efficiency and security of the entire VEC network. For example, malicious neighboring vehicles could intentionally drop or alter the data packets while serving as the forwarding nodes. Similarly, the malicious base station could leak the private information of vehicles or users in the data analysis. Thus, security and privacy issues must be taken into account for ensuring the operation and management of VEC networks.

2.2 Named Data Networking for VEC

NDN is one of the emerging information-centric network architectures. Compared with the current TCP/IP architecture, NDN has two major features: name-centric routing and addressing and a widely used in-network cache. The primary content transferred in NDN consists of *Interest* packets and *Data* packets. A *consumer*, who is interested in retrieving specific data contents, first sends interest requests carrying the *name* of the data to the NDN networks. The edge, intermediate, and core routing nodes in NDN forward the interest request based on the routing policies and rules until one of the routing nodes has the requested data in the local cache or the original data *producer*. The data packets are forwarded back to the consumer hop-by-hop along the reverse path as the interest packet, and more importantly, are cached by all intermediate routing nodes for serving the future requests of the same interests. Thus such an in-network caching feature is an efficient mechanism for improving the content distribution in NDN (Table 1).

Powering vehicular edge computing networks with the NDN architecture has a variety of unique advantages. First, NDN builds the naming mechanism to map the content resources and devices with human-readable semantic information, which is very critical for vehicles to retrieve security and traffic updates and discovery local services in a certain area at a certain time with different granularities [13, 34]. Second, the inherent in-network caching principle in NDN accelerates the dissemination of the traffic information

between vehicles [34, 43], and such rapid content distribution is often desirable for transmitting accurate and timely traffic congestion in major metropolitan areas around the world. Last, but most importantly, the NDN's ability to retrieve data based on interest, rather than IP-based end hosts, allow VEC nodes to switch different vehicles under high mobility scenarios or frequent node failures [38, 42]. When the vehicles communicate with the roadside infrastructure in its movement, it will face the problem that the current access point is not available and has to establish a connection with a new access point. Although some work focus on using technologies like forecasting next access point to realize seamless connection switch, the communication delays came with the mobility is always impossible to ignore, while the name-based data retrieval and extensive caching of NDN eliminate the dependency on static locations and connections.

These unique advantages make NDN a natural fit for VEC data communications, and some research work has explored how to construct the Internet of vehicles based on NDN. However, the challenges of data security and privacy issues remain in NDN-based vehicular edge computing networking. The NDN network without end-to-end connection needs an identity management mechanism to build trust, but unfortunately, the widely used identity-key-binding mechanism suffers from the problem that key authentication relies on a centralized root key, which is not suitable for an open P2P vehicle network. At the same time, the intermediate forwarding routers have the option of skipping the signature validation of data packets due

Table 1 The paraphrase of major nouns

Proper Name	Paraphrase
Vehicle Edge Computing(VEC)	Combining V2X with edge computing, VEC focuses on communication and computing collaboration between vehicles on the edge of the network and other vehicles or infrastructures
Vehicle to Everything(V2X)	Compared to vehicle-to-vehicle communication or vehicle communicating with roadside infrastructure, V2X enables vehicle-to-vehicle, vehicle-to-base station and even vehicle-to-anything communication, which facilitates vehicles to obtain a series of traffic information
Named Data Networking(NDN)	A new network architecture which uses names for content retrieval, routing and forwarding and improves the efficiency of content distribution by router cache
Blockchain	Blockchain is essentially a decentralized database with tamper-proof, traceable features, where each node adds content to the ledger through a consensus algorithm and keeps a full copy of the ledger
Consensus	Consensus algorithm is a protocol used to reach consensus among the nodes of blockchain

to the computation overhead, which opens the door for the attackers to inject forged or tampered data contents into the NDN networks and cause significant damage to vehicle nodes. Moreover, the semantic naming mechanism in NDN creates challenges for protecting the privacy of vehicles and drivers, which must be carefully addressed in the design of VEC networks.

2.3 Blockchain and Consensus Algorithms

Blockchain is an open and distributed digital ledger for effectively recording transactions between parties in a verifiable and untamable manner. In contrast to the traditional centralized ledger systems, all the participating nodes in the blockchain system maintain the distributed ledger collectively through the consensus algorithm. The blocks in the blockchain have a strict specification on their data structure where all the block headers are linked or chained through the cryptographic hash values of their prior blocks for ensuring the integrity of historical transaction records.

The consensus algorithm is a key component of the blockchain system for establishing the reliability and trust among participating nodes in such a distributed and decentralized computing environment without a trusted central authority. The core idea of the evidence-based consensus algorithms such as the Proof of Work (POW) mechanism of Bitcoin and the Delegated Proof of Stake (DPOS) mechanism of Ethereum. However, these evidence-based algorithms are often computationally expensive. As a result, the union or private chains prefer lightweight vote-based consensus algorithms such as the practical Byzantine Fault-Tolerant (PBFT) algorithm. In the PBFT consensus algorithm, the master node first broadcasts the message to all other nodes, which in turn verify and send votes to each other to reach an agreement. This design can tolerate up to one third Byzantine faults. However, this consensus mechanism, albeit computationally efficient, is very complex to implement, and becomes increasingly challenging to manage as the growth of participating nodes.

When blockchain is proposed as a solution, performance is always a concern as it is strongly tied to its consensus protocol design and hard-coded limitations on computations per block[36]. However, this is also a problem that scholars pay attention to and begin to solve. Some more efficient consensus algorithms are

proposed[22]. Furthermore, some research teams have made various optimizations for the query efficiency of blockchain, and proposed to improve the data structure[26, 29], or learn from the idea of efficient database[23] to achieve efficient query processing.

In recent years, a few research effort have been made to introduce the blockchain system into the NDN architecture such as BlockNDN [17] and BoNDN [4]. These earlier studies mostly focus on how to better integrate NDN and blockchain and how to design the forwarding and synchronization mechanisms for supporting all the basic and fundamental functions of NDN. However, there are few attempts considering how to design efficient consensus algorithms and how to explore the blockchain system for improving the security and privacy in NDN (Table 1).

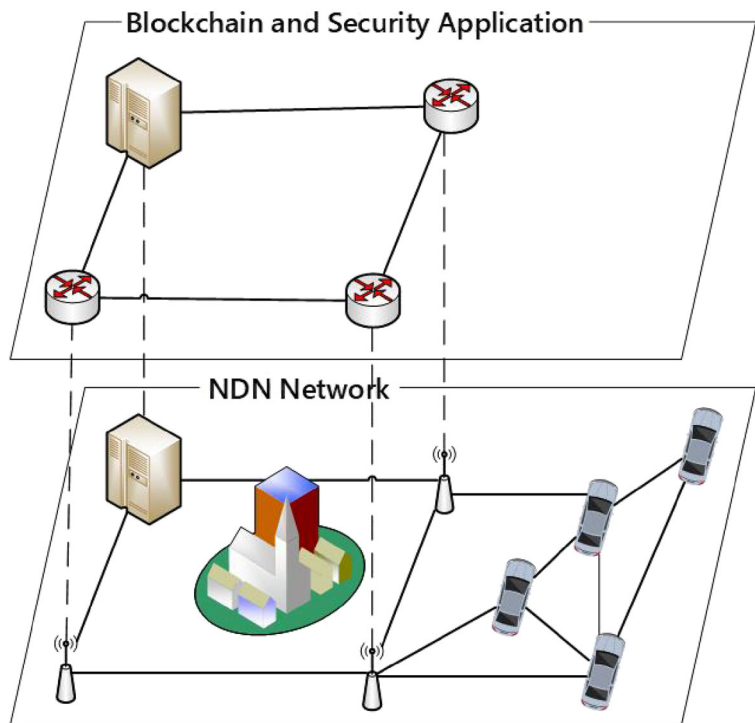
3 System Design Overview

In this section, we first describe the system model of our proposed blockchain-based security architecture for NDN-VEC networks. Subsequently, we present a simple yet effective delegate consensus algorithm for the blockchain architecture and discuss the design of the naming rules in the proposed system.

3.1 System Model

We consider a vehicle edge network where nodes including vehicles, base stations, and edge cloud servers communicate with each other via NDN. Given that vehicles act as data producers and consumers, they can inform their locations, speed and traffic information to nearby nodes including vehicles, base stations, and edge cloud servers in the area. Reciprocally, vehicles can also request similar location information and related traffic services from nearby nodes. Thus, the node identifier is supposed to identify and manage easily. For the V2V architecture, in addition to the base stations, the vehicles themselves also undertake the work on certain packet forwarding. Thus, it is more crucial to be free of cache poisoning in the data cache for secure and efficient forwarding in NDN based VEC networks. In addition, edge cloud servers play the role of the service provider and data producer in NDN. They collect and store information, or perform data analysis to provide various services.

Fig. 1 NDN-VEC model: A vehicle edge network where nodes including vehicles, base stations, and edge cloud servers communicate with each other via the NDN. The vehicles act as data producers and consumers, which can inform their locations, speed and traffic information to nearby nodes in the area. The edge cloud servers play the role of the service provider and data producer, which collect and store information, or perform data analysis to provide various services. Considering resource capacity, blockchain and its supported security services are deployed on base stations and cloud servers, while vehicles obtain security services via communicating with base stations



To allow producers to control their own data and finally guarantee data privacy, we propose the NDN-VEC system model, as illustrated in Fig. 1. Considering that resource capacity, blockchain and its supported security services are deployed on base stations and cloud servers, while vehicles obtain security services via communicating with base stations. Moreover, due to the overhead of the blockchain consensus described in Section 3.2, the base stations act as the ordinary blockchain node to submit and query transactions, while the cloud servers and the nodes with stronger storage and computing capacity serve as delegates to perform signature verification.

3.2 Delegate Consensus Algorithm

Considering a scenario with a large number of vehicle nodes as well as high-performance expectations, none of the existing blockchain consensus is a straightforward solution. For example, Proof of Work (PoW) and Proof of Stake (PoS) have good scalability and mature implementation, but they are criticized by poor performance. Byzantine Fault-Tolerant (BFT) protocol performs well in terms of efficiency, but sacrifices scalability due to complex communication, resulting

in its limitation on the network scalability. To this end, we propose the delegate consensus algorithm (DCA) based on delegation.

Inspired by delegated PoS (DPoS) and BFT, DCA achieves scalability and high performance at the same time. In DCA, a fixed number of delegation members are responsible to pack and verify blocks. Each individual member is dynamically elected and is allowed to generate blocks during a fixed time. A candidate block is set as valid only when it is confirmed by more than 2/3 of the delegates. The valid block is then propagated and verified across the whole network.

Specifically, DCA elects M delegates that serve as trust domain administrators, according to the actual scenario. Assume the time slice is fixed as δ_t , and each delegate solves an id based on the current *timestamp* for block generation as:

$$D_{id} = (\text{timestamp} / \delta_t) \bmod M$$

If D_0 is the current delegate to create blocks, it is now allowed to serialize a batch of transactions into a new candidate block B_{uv} and send it to all other delegation nodes. Upon receiving B_{uv} , each delegate verifies the validity of the block, including transaction verification and block verification. Transaction verification

varies from different applications while block verification focuses on block generation turn and packet signature. If the block is valid, the delegate node signs B_{uv} and send it back to D_0 . When D_0 receives more than $2 \times \frac{M-1}{3}$ valid signature from other delegates, it generates a multi-signature and adds it to B_{uv} to create a verified and valid block B_v . After this, the new block is broadcast and confirmed across the entire network. The future block productions continue with a similar process.

3.3 Naming Strategy in the System

The routing and forwarding of NDN are all based on the name, so naming is a significant part to be considered when using the NDN network. However, NDN leaves the design of namespace up to the developers and consumers to enhance the density of mappings between applications and their networks. NDN simply recommends that naming be structured in a hierarchical way, which helps to represent the relationship between content and name elements and helps to semantically represent contents.

When designing the naming rules in the proposed system, there are two significant considerations: easy-to-understand hierarchical naming and convenient-to-check naming. A clear naming strategy implies not only application functionalities but also attributes which are associated with data content, such as block height and block producer ID.

There are three parts of each name in the blockchain system: i) application name prefix, ii) function name prefix, and iii) block information prefix, *e.g.*

/application/function/blockinfo

Three functions will be defined and distinguished under the function prefix of broadcasting synchronous data for a blockchain system. The first function identifies specific block data, naming each block with block information prefix including block height, producer id, and block type. The second synchronization function is used to retrieve synchronization information from the network when the node is missing block information to restore the latest blockchain state, while the third function informs the other nodes of newly generated block information.

4 Security Applications and Analysis

The crux of improving security and privacy in NDN VEC is to ensure the authenticity, integrity and consistency of the data. The authenticity and integrity of data packages are guaranteed by digital signatures, and the verification of digital signatures depends on the authenticity of the corresponding key. In the scenario where multiple sites issue keys like VEC, it is worth paying attention to how to verify the authenticity of the key across multiple trust domains. Meanwhile, content poisoning attacks can damage the authenticity and integrity of data, so how to prevent content poisoning attacks without too much verification burden on nodes needs to be explored. Also, in order to ensure the consistency of the data, it is necessary to ensure that the data in the cache can also be controlled.

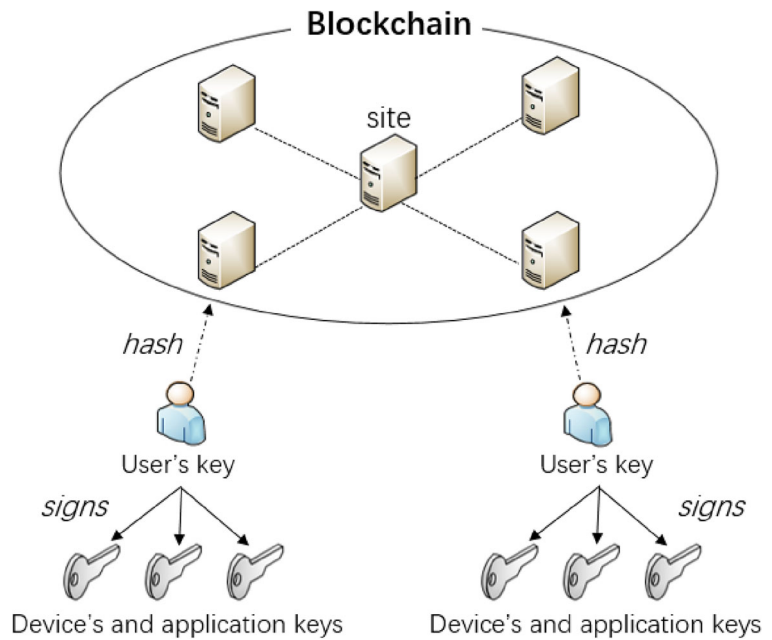
In this section, we describe the security applications of the proposed blockchain architecture for NDN-VEC networks in key management, cache poisoning attack detection, and access control. In addition, we analyze the security of the blockchain system based on the DCA consensus algorithm, and its guarantee on the data consistency.

4.1 Key Management

All systems encounter the common security problem of data authenticity and integrity, as the data that flow through the whole network are possible to be tampered, falsified, or attacked. In general, the public-private key technology is a straightforward solution, where the private key is used to sign data while the public key enables others to verify the signed data based on a certain trust rules. However, this type of scheme faces a dilemma in key management of how to gain trust from the keys, *i.e.*, how to guarantee key authenticity.

To enhance the reliability of key management, we propose a blockchain-based solution to endow key trust instead of a root key. In our design, key management is abstracted into three layers as shown in Fig. 2. The user is placed in the second layer in our design, and it will be responsible for managing all the devices and applications in its domain, such as the user's vehicle and the application for obtaining road traffic information. Like in the existing schemes, a user-level key has the right to give a trusted certificate of keys in the first layer by signing these keys, resulting in the

Fig. 2 Blockchain-based key management model: The user will be responsible for signing managing all the keys of devices and applications, such as the user's vehicle and the application for obtaining road traffic information. When verifying the authenticity of the device's and application key, we first trace back to the user who issued it, and then query the blockchain to verify whether information of the user's key is true



only way to verify the keys in the first layer is to trace back the user layer. The third layer is the blockchain layer, in which each site node stores the public keys of the user layer and the namespace that the users apply for authorization. The introduction of blockchain can not only store the relevant information of users' keys, but also be naturally suitable for verifying the authenticity of the key across the user trust domain.

The mapping of the user layer's public key and the device layer namespace it has authority over will be stored on the blockchain, and the key authenticity can be verified layer by layer by the following steps. Consumers get the validation information of the device layer's key through the field *KeyLocator* in the data packets, in which they will find the user-layer key itself and user-layer *key name*. The *key name* is used to check the authenticity of the user-level public key, and the process of check is simple: the consumer calculates the hash value of the user's public key using the SHA-256 function and compares it to the hash value of the user's public key stored in the blockchain to check if it is consistent. For quick lookup, *BlockHeight* and *TransactionHash* of the key authentication transaction will also be recorded in the public key packet.

4.2 Content Poisoning Attack Detection

The implementation of trust key management cannot guarantee the integrity and authenticity of data, because the intermediate router does not and can not perform the signature verification of packets due to limited resources. As a result, packets transmitted over the network can be poisoned, which can be divided into three categories:

- Corrupted content: the attackers simply tamper with the content of the packet and do not regenerate the signature based on the tampered data packet.
- Unauthentic content: the attackers tamper with the content and reconstruct the legitimate signature using his own key.
- Fake content: malicious producers use their own keys to sign and publish fake contents, such as wrapping a Trojan file as traffic information.

Our proposed blockchain-based security architecture employs a three-stage scheme to detect all categories of content poisoning attacks: retrieval stage, recovery stage, and feedback stage.

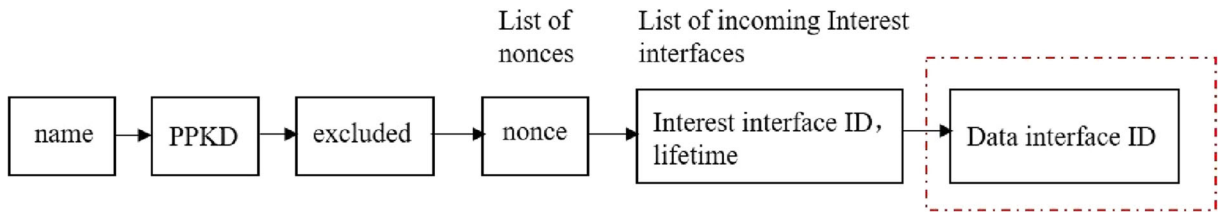


Fig. 3 PIT table items

4.2.1 Retrieval Stage

The primary purpose of the retrieval stage is to identify and eliminate untrue poisoned contents in a lightweight but efficient manner. The user is required to add self-authenticated information in the interest packet when requesting the content.

Specifically, the public key digests information of the content publisher (PPKD) corresponding to the requested content is written to the PPKD field of the interest packet. The intermediate node will verify whether the returned data packet matches the self-authentication information. In other words, the routing node will obtain the publisher’s public key (PK) from the KeyLocator field in the data packet, then compare the hash of the public key with PPKD in the corresponding table of PIT, and discard the data packet if:

$$\text{hash}(\text{PK}) \neq \text{PPKD}$$

Besides, the information of the PPKD domain will also be recorded in the PIT table, whose items in this scheme also have an additional record of the data face (DF), so that the source interface of poisoned content can be found in the recovery stage and the forwarding

state of the interface can be adjusted accordingly, as illustrated in Fig. 3. If the validation fails, the corresponding FIB table forwarding status is modified and marked *Yellow*. According to the forwarding priority, the forwarding status in the FIB table can be successively divided into *Green*, *Yellow*, and *Red*. When the forwarding status is *Red*, it means that the interface cannot work normally, and the forwarding to the interface will be prohibited.

4.2.2 Recovery Stage

The corrupted contents can not be identified by intermediate nodes because of lacking signature verification. Thus the content requester identifies the corrupted contents and sends a Feedback Interest Packet (FIP) with excluding filters to get the real content. As shown in Fig. 4, exclude filter in the FIP indicates the hash value of the poison packets received by the user, and a complete poison packet is attached to the FIP.

Depending on the exclude filter field in the FIP, each router will check whether the poisoned content is cached in the local CS, and, if found, deletes the

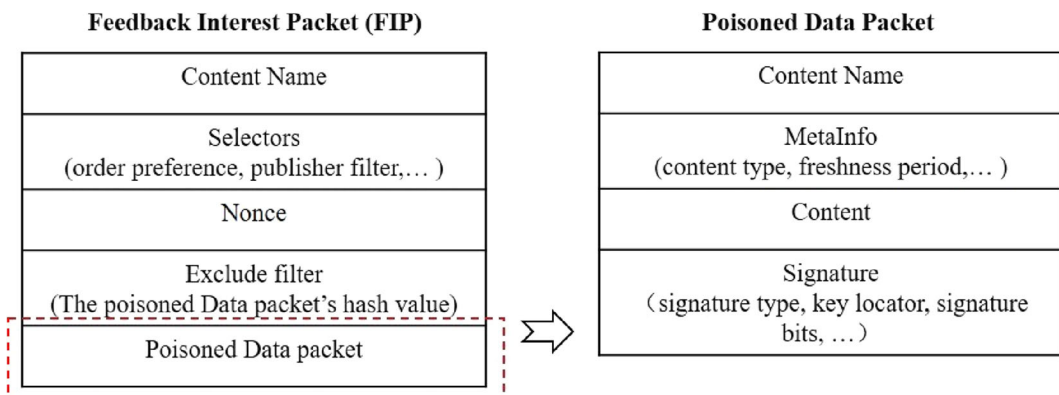


Fig. 4 Feedback interest packets

packet and adjusts the forward policy by marking the forwarding status of the correlative interface as yellow. In order to prevent content poisoning again and speed up the acquisition of real content, the intermediate routing node will determine whether the interface of packet arrival is green or not and if it is not, it will actively do signature verification. If the signature is verified successfully, the intermediate routing node removes the PIT entry with the DF record and changes the FIB forwarding status back to Green. Otherwise, it can be inferred that the last hop of the data packet is a contaminated router or content server, and the node will directly discard the data packet and modify the FIB forwarding status of the return interface to red.

4.2.3 Feedback Stage

The last stage is designed to identify fake contents and potential malicious content publishers. After receiving the false content from the malicious content publisher, the content requester can send back *Feedback* transaction to the blockchain, which means that the user accuses the content publisher whose public key hash is PPKD. If a content publisher was issued by multiple users at the same time, it is more likely a malicious content publisher.

When the number of transactions accusing a certain node exceeds a threshold, a *decision* transaction will be triggered, which requires the consensus node to actively verify the validity of the content. The *decision* transaction confirmed by consensus represents the consistent confirmation of poisoned content by the blockchain nodes, so the reputation of the content producer which influences the right of publishing content will be reduced.

4.3 Access Control

The key management mechanism and measures to prevent content poisoning attacks can only guarantee the authenticity and integrity of the data in the process of delivery. One thing that also should be ensured is that the subject receiving the data has the right to view it, which is guaranteed by the access control mechanism.

In this paper, the symmetric encryption algorithm is adopted to encrypt the content with a symmetric data key, and access control is implemented by controlling the distribution of data key, so that it

can only be obtained by authorized consumers. This design avoids the problem of cached copies in the routers that do not perform the same access control policy, because the cached copies now are also encrypted. In terms of the implementation of access control policies, this paper integrates the attribute-based access control, names the access policies hierarchically, proposes a distributed access control model based on blockchain, and designs a complete access control process on this basis. Attribute-based access control abstracts real identities into attributes, which can describe access control policies in a fine-grained way. The consensus mechanism of blockchain can introduce distributed policy decision points, which can avoid single point failure of central decision points and help realize fine-grained access policy decisions. On the other hand, the hierarchical semantic naming strategy in NDN has the function of describing access policies in a fine-grained way.

4.3.1 Attribute-based Access Control Strategy

Compared to other access control mechanisms for NDN-VEC networks, this paper proposes to adjust the attribute-based access control mechanism so that the access control decisions do not have to rely on a centralized policy decision point. Instead, the subject requesting access certifies to the entire blockchain that satisfies the resource access policy set by the resource owner. On the other hand, a hierarchical semantic naming scheme is still used because it allows fine-grained descriptions of access policies. The following part will describe the blockchain-based access control mechanism in detail.

Inspired by the attribute-based access control model (ABAC), our proposed access control strategy uses a set of attributes to represent the resource and the subject requesting resource. In addition, we explore blockchain to record the access control policy made by the resource owner as well as determine whether the subject meets the access policy for obtaining the resource. Finally, the subject will publish a transaction to prove to all consensus nodes that it satisfies the access control policy.

4.3.2 Transactions in Blockchain

- *Register*: when the subject wants to access a particular resource, the subject it publishes a registered transaction to the blockchain. The input

of the transaction is the subject’s public key, and the output is the name of the resource it wants to access.

$$T_{register} = (Input(SPK), Output(ResourceName))$$

- *Authorization*: an authorization transaction is a transaction in which the resource owner informs the subject of access policy. The resource owner will build the access control policy according to the attribute set provided by the attribute agency, then take the public key of the access control policy (policy public key, PPK) and the random number encrypted by the subject’s public key as the output of the transaction. It should be noted that the access strategy built by the resource owner is converted by the system into a CP-ABE access tree, and only the subject that satisfies the access tree attribute can decrypt the source data encrypted by this access tree. The PPK stored in the authorization transaction points to the access tree for easy access by subsequent subjects. The input of this transaction will be the name of the resource requested for access and the signature signed by the public key of the resource owner. This is done to facilitate tracing back to the registered transaction for which the principal requested access to the resource.

$$T_{authorization} = (Input(ResourceName, Sig_{ropk}), Output(Nonce_{spk}, PPK))$$

- *Access*: an access transaction can be viewed as a process by which the subject proves that it satisfies the access policy of the resource owner in the blockchain. When the subject queries the blockchain for an authorization transaction related to its request, it uses the PPK provided in the transaction to find the access control strategy of the resource owner, that is, a CP-ABE access tree. The subject then attempts to decrypt the access tree, and if it satisfies the access control policy, a key from the access tree (Attribute Secret Key, ASK) can be easily decrypted. The subject will use ASK as the input of the access transaction to prove that it satisfies the access control policy.

Whether the ASK submitted by the subject in the access transaction is an ASK encrypted by the resource owner using the access tree will be verified by multiple consensus nodes through the consensus process. The consensus nodes will obtain

their property from the property agency through the principal’s public key and use the property to decrypt the access tree, then they check whether they can get the ASK or the consistency of the ASK they decrypted and recorded in the access transaction. If verified, the access transaction will be successfully recorded on the blockchain, which proves that the subject has access to the resource.

$$T_{access} = Input(Sig_{spk}), Output(ROPK))$$

- *Revoke*: when the resource owner wants to revoke the access control policy that was previously notified to the subject, it is supposed to publish the revoke transaction. The input is the subject public key, and the output is an empty set and the public key of the previously published access control policy, representing replacing the policy with an empty set.

$$T_{revoke} = (Input(SPK), Output(\emptyset, PPK))$$

- *Feedback*: feedback transactions are designed to identify resource owners in the network who are maliciously denying access. When the resource owner queries the access transaction related to the subject recorded in the blockchain, that is, the subject has access to the resource, it should give the subject the required resource. In the case that for some reason the resource owner still refuses access, the principal registers the denial on the blockchain as a feedback transaction. The transaction input is the public key of the subject (SPK) and the output is the public key of the resource owner (ROPK).

$$T_{feedback} = (Input(SPK), Output(ROPK))$$

A simple threshold is used to prevent maliciously denied access and maliciously marked resource owners from denying access. Use n to count access transactions associated with a resource owner, and N to count represent feedback transactions associated with that resource owner. If the rejection coefficient $\mu = \frac{n}{N}$ exceeds a certain threshold, the resource owner will be observed on a blacklist, which will be published via the blockchain. Similarly, let n_2 represents the registered transaction related to the specific subject, and N_2 represents the feedback transaction

related to the subject. If the feedback coefficient $\rho = \frac{n_2}{N_2}$ is higher than a certain threshold, the subject is considered to have maliciously marked the resource owner.

4.3.3 Resource Access Steps

When the subject wants to access a particular resource, the whole process of access, as shown in Fig. 5 can be summarized into three steps, first, the subject send information to the resource owner, showing the application of resources, in the second place step, the subject issues a series of transactions to the blockchain to prove to the blockchain that it satisfies the access control strategy, and third, the subject obtains resources or feeds back to the blockchain of denying access.

The first step can be summarized as a subject sending an interest packet to the resource owner indicating which resource it is interesting. The interest packet will contain the subject public key (SPK) that identifies itself and a data key name that identifies the interest resource. For example, “*car/PKU/A1/14-15UTC/loca*” represents the location of the car of Peking University marked as A1 at 14-15UTC.

Subsequently, the subject proves to the blockchain that it satisfies the access policy. It will first send the register transaction to declare its request, then the resource owner issues the authorization transaction to indicate the access control policy, and finally, the subject issues the access transaction to submit its own proof of meeting the policy, in which the consensus node determines whether the proof is valid for the chain. The details are explained in the *Access transaction*.

The last step is to obtain resources and submit feedback feedbacks. After the access transaction is successfully confirmed by the consensus node, the subject sends an access packet to the resource owner to remind the resource owner that it satisfies the access policy. The access packet contains two items, the first of which is a random number of *Nonce*. This random number is stored in the authorization transaction created by the resource owner and encrypted by the Subject Public Key. The subject uses its private key to decrypt the value of the random number to prove to the data owner that it is indeed the owner of the public key. The resource owner verifies the random number and access transaction, and then sends the resource to the principal with its specific name.

4.4 Security Analysis

When introducing blockchain into our architecture, it is a natural question whether the security of the blockchain itself has an impact on the overall scheme. Many of the security issues of the blockchain are related to its consensus algorithm. For example, 51% Attack is directed to the POW algorithm, and the BFT algorithms face the threat of sybil attacks and eclipse attacks. Participants in the blockchain are concerned about whether they can reach an agreement safely, and the resulting blockchain will not fork or appear abnormal.

Now we analyze the security of the blockchain system based on the DCA consensus algorithm and its guarantee on data consistency via the following theoretical analysis and proofs.

Proof Assuming that the number of delegate members is M , as long as more than $2M/3 + 1$ delegates do not fail, the production of block B_v is legal and irreversible.

DCA algorithm requires the delegate group to verify the unverified block B_{uv} and send signatures to its producer. Only when more than $2M/3$ delegate's signatures are valid can a legal B_v block be generated, so a legal B_v block represents that more than $2M/3+1$ delegates have approved the B_{uv} block in this round. There are no more than $M - (2M/3 + 1) = M/3$ delegates failing at most, so the illegal B_{uv} block can only receive $M/3$ confirmations at most, and cannot become a legal block. Even if delegate in this round send different B_{uv} block to different delegate set, as long as one B_{uv} block received more than $2M/3 + 1$ delegate's confirmation, which means at least $M/3 + 1$ honest delegate confirmed B_{uv} , another block can only receive confirmation from $M - (M/3 + 1) = 2M/3$ delegates because the honest delegate cannot sign two different blocks in the same round. Therefore, as long as more than $2M/3 + 1$ agents do not fail, the production of block B_v is legal and irreversible. \square

Proof Legitimate transactions can eventually be packaged into blocks and confirmed by the blockchain. DCA algorithm requires a set of delegates to generate blocks in turn. Although the dishonest delegates will send the illegal block or not produce the block, the previous proof guarantees that the illegal block cannot be confirmed, and the next delegate will still package the transaction and produce the block as long as an

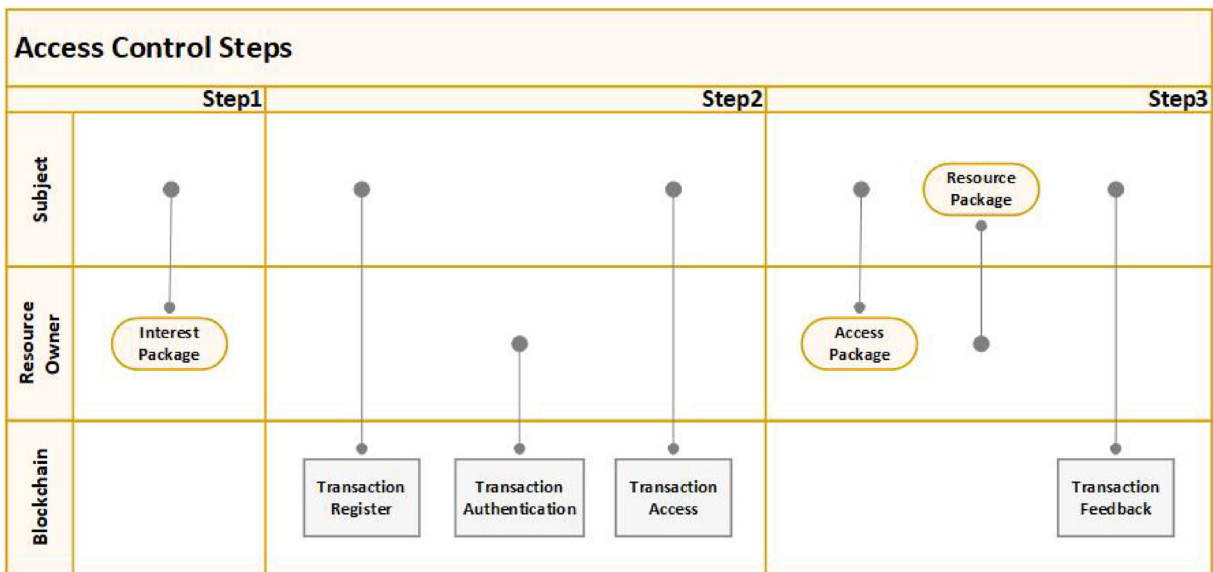


Fig. 5 The overall process of resource access: A subject sending an interest packet to the resource owner indicating which resource it is interesting. Then the subject proving to the blockchain that it satisfies the access policy, including

subject sending the register transaction, resource owner issuing the authorization transaction and subject issuing the access transaction. Finally the subject obtain resources and submit feedback

honest working delegate exists. Therefore, legitimate transactions can eventually be packaged into blocks and confirmed by the blockchain. □

5 Experimental Evaluations

In this section, we first examine the performance of the proposed blockchain security architecture for NDN-VEC networks. Subsequently, we present the performance and cost of the proposed architecture in detecting and mitigating content poisoning attacks under two different types of VEC network topology. We conclude this section by presenting the overhead of the proposed access control scheme.

5.1 Blockchain System Evaluations

We implemented a prototype blockchainv system over NDN via Networking Forwarding Daemon (NFD). The system consists of the core components including DCA. In addition, the ndnSIM platform, a NS-3 based Named Data Networking (NDN) simulator provide an extensive collection of interfaces and helps perform detailed tracing behavior of every component, is used to simulate the content poisoning mitigating and access control, and we evaluate the performance

and cost of both schemes on local machines with Intel Core i5-6500, 3.2GHz and 8GB RAM.

The performance of a blockchain system is mainly evaluated by two metrics:

- **throughput:** the number of transactions per second processed by the blockchain system (TPS)
- **delay:** the time of a transaction from its initiation to its final confirmation.

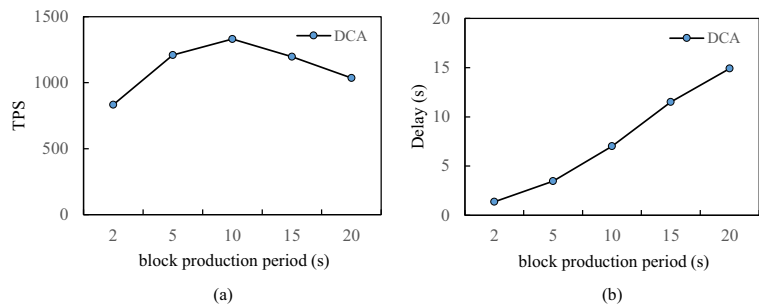
The time delay includes the transaction propagation time $T_{Broadcast_{tx}}$, the block propagation time $T_{Broadcast_{block}}$, and the consensus time $T_{Consensus}$. In summary, the delay metric is calculated as

$$Delay = T_{Broadcast_{tx}} + T_{Consensus} + T_{Broadcast_{block}}$$

Higher TPS and lower latency are essential for VEC networks, as there are thousands of vehicles constantly taking advantage of blockchain security applications. Our extensive experiments lead to the following observations:

1. TPS changes with different block interval: As shown in Fig. 6a, the TPS of the DCA algorithm changes with block interval. With the increase of block frequency, the TPS of the algorithm increases. At the block interval of 10 seconds,

Fig. 6 Blockchain system performance: **a** TPS on different block production period; **b** Delay on different block production period



DCA can achieve the highest throughput of 1,331. When the interval of producing block increases to 15 seconds and 20 seconds, the algorithm TPS starts to decline slightly, which is due to the fact that the block interval is related to the block size. The larger the block interval is, the more transactions can be accommodated in a block. In another words, the more transactions can be processed during a unit time. However, the large block size causes longer broadcast delay, and agents who do not work properly will not be able to generate valid blocks in its generation round. Therefore, the larger the block interval, the greater the probability that there is no block generation for a long time. Figure 7 illustrates TPS metrics of several common algorithms or platforms. The PoW in Bitcoin only handles 7 TPS on average, with blocks size about 1M. The TPS in Ethereum is also unsatisfactory. The BitShare platform achieves 500 TPS via DPoS mechanism. Compared with these mainstream consensus of blockchain, the delegate consensus algorithm adopted in this paper can achieve higher throughput with nearly **1,350** TPS in the underlying network architecture of NDN.

- Delay changes with different block interval: As shown in Fig. 6 (b), the delay of DCA linearly increases with the increase of block interval. The increase of block interval increases the time between the transaction generation and its packet. Moreover, the delay of broadcast block $T_{Broadcast_{block}}$ also increases due to the increased transaction volume contained in a block. As the total time to verify a transaction signature increases, $T_{Consensus}$ will increase correspondingly. Therefore, the increase in block time interval will lead to an increase in delay.

5.2 Evaluation of Content Poisoning Detection and Mitigation Mechanism

We conduct experiments under both simple and complex network topologies as shown in Table 2. Compared with IKB based schemes and multi-channel forwarding and schemes based on signature verification, our solution aims to receive the response of interest packets that regards traffic conditions and other information within the shortest delay. Towards this end, we mainly evaluate three major metrics: number of interest sent, number of signature verification, and request delay:

- Number of interest sent:** for each content acquisition, the less the average number of interest sent, the less the number of trips back and forth representing interest, and the number of packet

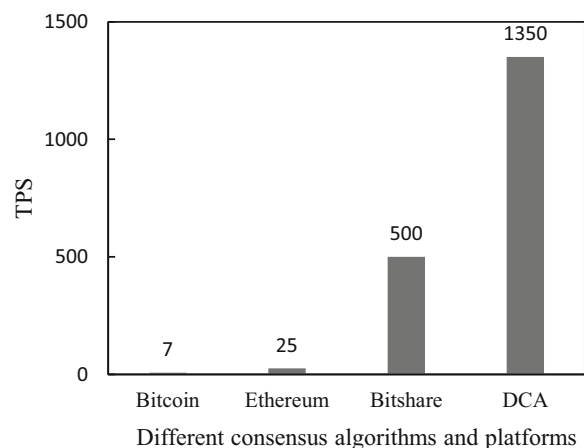


Fig. 7 TPS performance metrics of different consensus algorithms and platforms

Table 2 Experimental parameter setting

Parameter	Simple topology	Complex topology
Link bandwidth (Mbps)	10	10
Link delay (ms)	10	10
Number of routers	4	30
Number of consumers	1	16
Number of producers	1	1
Number of attackers	1	2
Data size (byte)	1024	1024
Number of content items	100	100
Interest rate(content/s)	20	20
Zipf parameter (s)	1.0	1.0
CS size(units)	20	40
Caching probability	0.7	0.7
Cache replacement policy	LRU,LFU,FIFO,Random	LRU
Simulation time(s)	10	10

signature verification will be less, which is critical to the average latency of the request.

- **Number of signature verification:** it is costly for vehicles to act as the intermediate router to perform validation. Thus, the higher is the number of signature verification, the longer is the request latency.
- **Request delay:** this metric is the key to evaluate the efficiency of our content defense scheme. Even if there are attackers and content poisoning in the network, the smaller the request delay is, the faster the consumer vehicles can retrieve the

real content and the faster the intermediate router vehicles can eliminate the poisoned content.

5.2.1 Simple Network Topology

For cache replacement strategies including LRU (Least Recently Used), LFU (Least Frequently Used), FIFO (First Input First Output), and Random, we evaluate the average number of times that users send interest packets under three schemes, respectively. Fig. 8(a) presents that our method and IKB scheme results in the same for different cache replacement strategies

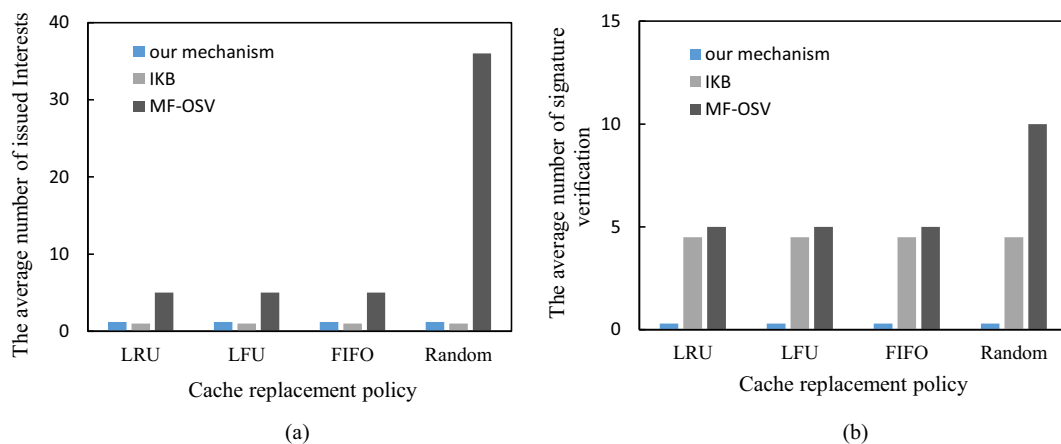


Fig. 8 **a** The average number of issued Interests on different cache replacement policy; **b** The average number of signature verification on different cache replacement policy

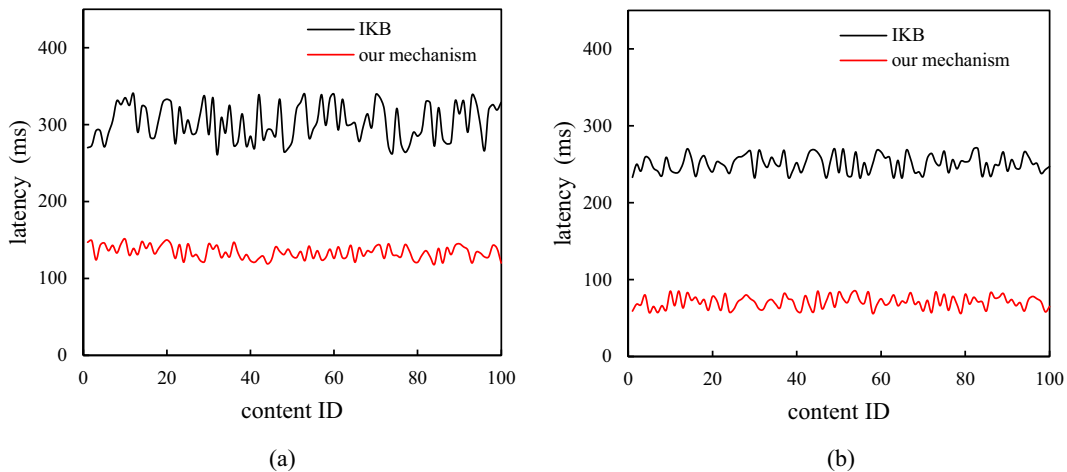


Fig. 9 **a** The latency result when the attacker disseminates corrupted contents; **b** The latency result when the attacker disseminates unauthentic contents

with 1.2 and 1 respectively. As for mf-osv, the average number of interest packets sent under LRU, LFU and FIFO policies is 5, while under Random policy is 36. As shown in Fig. 8(b), our defense scheme requires the least number of signature verifications at only 0.3 on average. IKB requires 4.5 signature verification operations, while mf-osv requires an average of 5 signature verification operations under LRU, LFU and FIFO policies, even 10 signature verification operations under Random policy. In addition, the defense scheme in this paper only needs to perform signature verification in the recovery process, and only one consumer performs the verification. After receiving

FIP, the router will mark the interface forwarding status of the attacker to red according to DF information in the corresponding table of PIT. Therefore, our scheme achieves the least signature verification.

Figure 9(a) elaborates the average request delay when the attacker sends corrupted content. Our poison avoidance scheme based on blockchain and forwarding strategy reduces the delay of IKB by 56% with only 134 ms. Figure 9b shows that our scheme optimizes the request delay when the attacker sends unauthentic content. IKB solution's delay is 253 ms while our scheme is 71 ms. Although the IKB scheme has the small average number of sending interest

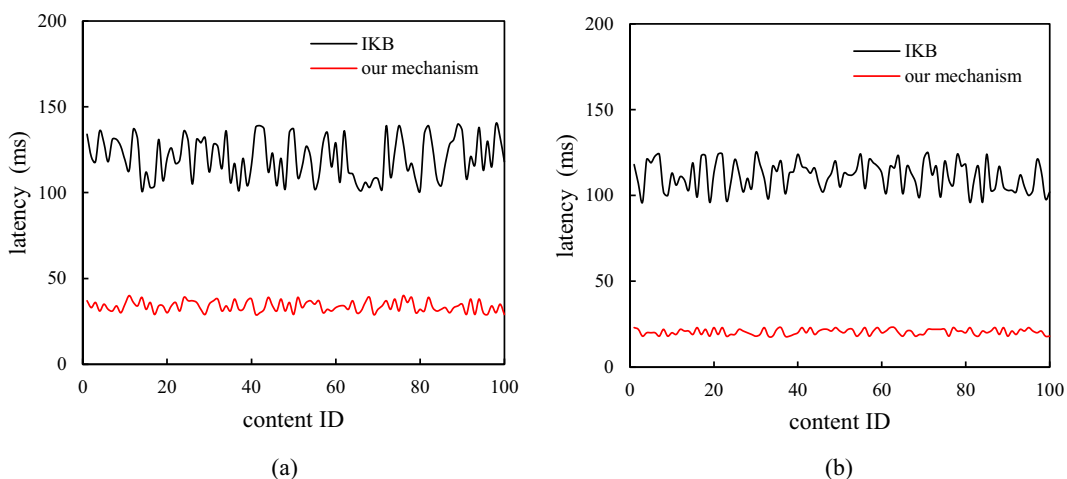


Fig. 10 **a** The latency result when the attacker disseminates corrupted contents; **b** The latency result when the attacker disseminates unauthentic contents

packets, it requires a lot of signature verification, and too many signature verification operations reduce the performance, causing a large average request delay.

5.2.2 Complex Network Topology

The complex topology simulates the network topology of Deutsches Forschungsnetz (DFN), also referred to as German Research Network, which includes 30 routers, 16 consumers, 1 legitimate content publisher, and 2 attackers. As shown in Fig. 10, the average request delay of the IKB scheme with two attacks is 121 ms and 112 ms, while the request delay of our scheme is 34 ms and 21 ms, both much lower than that in the IKB scheme. In complex network topology, the ratio of the cache hit increases, therefore the average request delay of the user decreases. Through experiments in simple and complex network topologies, it is proved that our defense scheme can effectively defend against content poisoning attacks, enable intermediate nodes to identify and eliminate poisoned data packets at low cost, and enable consumers to retrieve real content quickly.

5.3 Evaluation of Access Control Model

For access control, high TPS of the underlying blockchain supports high access frequency. This section tests the computation and storage overheads of the encryption algorithm in the access control system to see whether it is acceptable in the VEC scenario. In this paper, the CP-ABE encryption algorithm is adopted and data packets of four different sizes, i.e., **512 bits**, **1,024 bits**, **1,536 bits** and **2,048 bits**, are used to simulate data keys that are encrypted. The number of attributes is set to different values successively and the cost of key generation, encryption and decryption of different packets will be tested. The final cost is averaged for 4 tests.

5.3.1 Calculation Overhead

When the number of attributes is adjusted from 5 to 30, the time to generate the key is shown in Fig. 11a. Although the generation time of the decryption key generally presents an upward trend with the increase of the number of attributes, the difference between the generation time of 30 attributes and that of 5 attributes is about 0.07 seconds, so the generation time

of the key is almost irrelevant to the size of the data key. As Fig. 11b shows, the time required for encryption increases as the number of attributes increases, but the increasing trend is small. When the number of attributes is 30, the encryption time is less than 0.25 seconds, which increases by less than 0.2 seconds when the number of attributes is 5. Moreover, at the same number of attributes, there is little difference in the time required for the encryption between the four kinds of data packets, meeting the encryption time requirement for data key in VEC.

Figure 11c proves that the time required for decryption raises with the increase in the number of attributes, and this increase is even smaller than the increase in encryption. The difference in the time required for decryption between the number of attributes 5 and 30 is only about 0.05 seconds. When the number of attributes is 30, the maximum time required for decryption is slightly more than 0.07 seconds, which also meets the requirement of data key decryption efficiency in VEC.

5.3.2 Storage Overhead

The storage cost also reflects the feasibility of a framework. Using the CP-ABE algorithm to encrypt four kinds of data packets, and setting the number of attributes from 15 to 35 successively, the memory usage was measured for different attributes when each packet was encrypted. As shown in Fig. 12a, for the same packet, the storage footprint of different attribute numbers is almost the same. For example, for the packet at a size of 1,024 bits, when the number of attributes varies from 15 to 35, the memory usage remains at 300 bytes, indicating that the number of attributes has little difference in the storage overhead.

In the decryption process, as presented in Fig. 12b, the storage size occupied is related to the packet size while is almost not influenced by the number of attributes. For example, for the 502bit packet, the memory occupied by the number of five different attributes is between 100 bytes and 200 bytes.

In summary, our extensive experiments have demonstrated that the overhead of encryption and decryption can be accepted by the VEC scenario, and the changes brought by different attributes and packet sizes are small, which indicates that our scheme has good scalability for VEC.

6 Related Work

6.1 Security Application in NDN

As NDN starts the real-world deployment for mission-critical applications, the security and privacy have become the most serious issues in the NDN and cybersecurity research community [21, 25, 30, 33, 37]. However, few attempts have been made to systematically study key management issues in NDN.

Some efforts have introduced new certificate format [40] or new namespace design scheme [32] to simplify the process of identity authentication, while the work in [15] proposed the binding of the identity of the publisher, the public key of the publisher, and the name of data for ensuring the validity of the public key. Recent research in [41] has proposed a web-of-trust model to allow users to collaborate on identity authentication, which inspires our blockchain-based trust and key management scheme developed in this paper.

The widely adopted defense strategies against content poisoning attacks rely on intermediate routers in NDN for content verification. However, the studies in [2, 8] have revealed the weakness and overhead of signature verification by intermediate routers. The scheme developed in [10] binds the content naming prefix with publisher public key digest (PPKD), but requires non-trivial computational overhead on resource-constrained vehicles. Rather than relying on active content verification, [3] has proposed an alternative scheme to explore the passive feedback from the consumer to inform downstream nodes on the poisoned content. The feedback triggers the intermediate nodes to lower the ranking of the corresponding content, thus reducing the probability of content

poisoning by prioritizing highly ranked content. On the other hand, attackers may maliciously send a large amount of feedback data to exhaust the computing resources of the intermediate nodes to reduce the effectiveness of the feedback-based content poisoning attack scheme [18].

The access control issue in NDN has received significant attention due to the importance of contents in the NDN architecture. In general, the data encryption and signature in NDN have provided certain degree access control. However, such general access control schemes require strong security of key distribution [14], the high reliability of access administrators [44], and efficient use of the in-network caching feature in NDN [5]. Recently, an interest-based access control scheme [45] is proposed to leverage the forwarding feature in NDN where intermediate nodes make the forwarding decisions only based on the information of interest packet. These access control schemes incur substantial computation overhead for encryption/decryption operations, thus are not suitable for connected vehicles with limited computation resources [6, 16, 39]. Different from these prior studies, our research effort in this paper focuses on developing a blockchain-based and lightweight security framework for providing a variety of security services such as key management, cache poisoning attack detection, and access control in NDN-based VEC networks.

6.2 Security Application in TCP/IP Network

In TCP/IP networks, some works have discussed enhancing key management with blockchain technology. [1] proposed that the key management scheme based on a third-party key generation center (KGC)

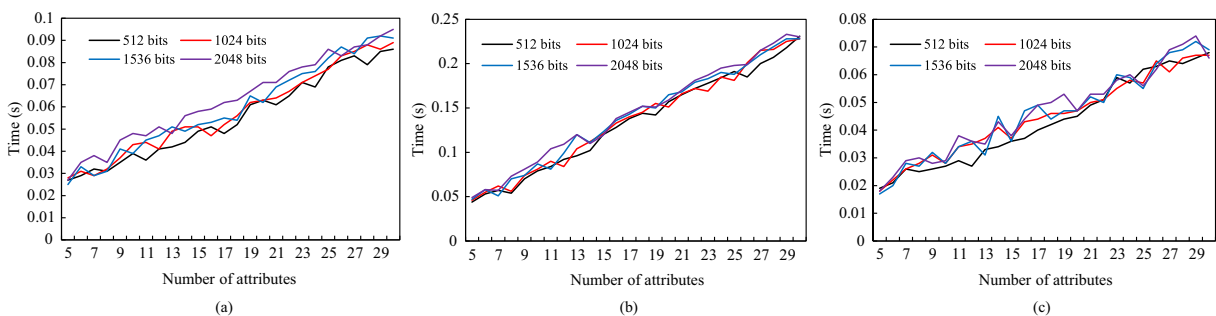


Fig. 11 **a** The impact of the number of attributes on the key generation time; **b** The impact of the number of attributes on the encryption completion time; **c** The impact of the number of attributes on the decryption completion time

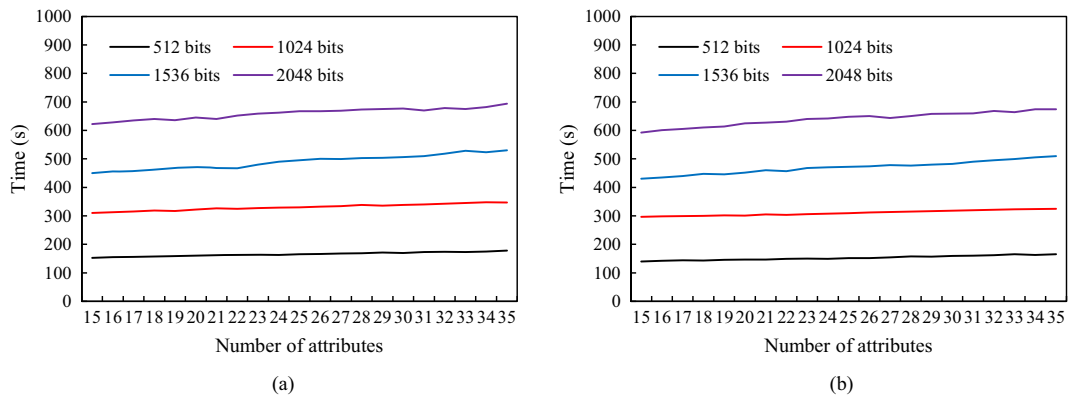


Fig. 12 **a** The impact of the number of attributes on the storage overhead required for data encryption; **b** The impact of the number of attributes on the storage overhead required to data decryption

or certificate authority (CA) is too trusting of the third party to be suitable for user-oriented privacy scenarios, while the blockchain eliminates the potential challenges posed by trusted third parties. In the view of [20], the introduction of blockchain can accelerate information transmission between security domains in the case of distributed existence of multiple security managers managing keys in their domains, because the information is transmitted directly between two security domains rather than through a central manager. Similarly, the purpose of introducing blockchain to the key management in this paper is solving the problem of single-point failure in key authentication and cross-domain authentication in distributed scenarios. The difference is that the first two schemes directly use the bitcoin blockchain, while the scheme in this paper improves the design of the underlying blockchain, such as a new consensus algorithm.

Content poisoning attacks are more common in the NDN network. In order to guarantee the authenticity of the content, the content in NDN must be signed by the producer on request, but for reasons of content distribution efficiency, the router does not be requested verifying signature compulsively, so an attacker may inject fake content into the router's cache, resulting in fake content being later assigned to the consumer[11]. In TCP/IP networks, a similar poisoning attack is the DNS cache poisoning attack. Specifically, the attacker can attack the URL name servers, replacing the correct IP address of the domain name in the cache with malicious server IP address, to provide an error response

later[24]. The defense against this kind of poisoning attack mainly requires the server to use ports randomly during the query to avoid the attacker to find the corresponding ports[19]. However, there are differences between DNS cache poisoning attacks and content poisoning attacks, as the implementation of DNS cache poisoning attacks requires the attacker to analyze the DNS server port while content poisoning attackers only need to intercept and modify data packages on the communication channel, which is inevitable. So the discussion of DNS cache poisoning attacks has limited significance to what we discuss in this paper.

Resource access control is a means to protect user privacy. In today's access control scheme, some work has focused on how to achieve lightweight access control in the scenario with resource-constrained devices such as the Internet of things. LACS[28] proposes a lightweight tag-based scheme that verifies the identity of IoT nodes by verifying the integrity of the shared files embedded with tag values for achieving access control. Some work has also used blockchain technology in the scheme. FairAccess[27] creatively utilizes issuing and using digital tokens for access control. However, due to the caching feature of NDN, whether the content cache copy performs the same access control policy should also be taken into consideration, which is not considered in the TCP/IP network access control scheme. The scheme proposed in this paper meets this requirement in the access policy verification.

7 Conclusions and Future Work

This paper introduced a blockchain-based security architecture for protecting the security and privacy for NDN VEC networks and developed three security mechanisms for key management, cache poisoning detection, and access control. We designed and implemented a novel and efficient blockchain system on NDN with a lightweight delegate consensus algorithm. Our extensive experimental evaluations based on the ndnSIM platform have demonstrated that the proposed blockchain architecture is suitable for VEC network scenarios. Specifically, the proposed key management scheme solves the trust problem caused by a single trust anchor, the cache poisoning detection method protects VEC networks from the cache poisoning attacks with lightweight computational and validation overhead on intermediate nodes, and the access control strategy supports fine-grained access control over VEC resources and validates access rights via the blockchain system.

Considering the blockchain's own mechanism and the security issues currently discussed, more detailed impact on overall efficiency and safety as well as the optimization scheme when introducing the blockchain in our architecture will be one of the key points of future work. Meanwhile, our future effort is centered on extending blockchain-based security framework for a variety of Internet of Things (IoT) network architectures such as NDN-based unmanned aerial vehicle ad hoc networks and smart building networks.

Acknowledgment This work was financially supported by Shenzhen Key Laboratory Project (ZDSYS201802051831427) and the project "PCL Future Regional Network Facilities for Large scale Experiments and Applications".

References

1. Lei, A., Cruickshank, H., Cao, Y., Asuquo, P., Ogah, C.P.A., Sun, Z.: Blockchain-based dynamic key management for heterogeneous intelligent transportation systems. *IEEE Internet of Things Journal* **4**(6), 1832–1843 (2017). <https://doi.org/10.1109/JIOT.2017.2740569>
2. Baugher, M., Davie, B., Narayanan, A., Oran, D.: Self-verifying names for read-only named data. In: *Proceedings of IEEE INFOCOM Workshops* (2012)
3. Cesar, G., Gene, T., Ersin, U.: Needle in a haystack: Mitigating content poisoning in named-data networking. In: *Proceedings of NDSS Workshop on Security of Emerging Networking Technologies (SENT)* (2014)
4. Chen, D., Zhao, H.: Data security and privacy protection issues in cloud computing. In: *Proceedings of International Conference on Computer Science and Electronics Engineering* (2012)
5. Chen, T., Lei, K., Xu, K.: An encryption and probability based access control model for named data networking. In: *Proceedings of IEEE International Performance Computing and Communications Conference (IPCCC)* (2014)
6. da Silva, R.S., Zorzo, S.D.: An access control mechanism to ensure privacy in named data networking using attribute-based encryption with immediate revocation of privileges. In: *Proceedings of IEEE Consumer Communications and Networking Conference (CCNC)* (2015)
7. Deng, M., Tian, H., Lyu, X.: Adaptive sequential offloading game for multi-cell mobile edge computing. In: *Proceedings of International Conference on Telecommunications (ICT)* (2016)
8. Gasti, P., Tsudik, G., Uzun, E., Zhang, L.: Dos and ddos in named data networking. In: *Proceedings of International Conference on Computer Communication and Networks (ICCCN)* (2013)
9. Gerla, M., Lee, E.K., Pau, G., Lee, U.: Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds. In: *Proceedings of IEEE world forum on internet of things (WF-IoT)* (2014)
10. Ghali, C., Tsudik, G., Uzun, E.: Network-layer trust in named-data networking. *ACM SIGCOMM Computer Communication Review* **44**(5), 12–19 (2014)
11. Ghali Cesar, T.G., Ersin, U.: Needle in a haystack: Mitigating content poisoning in named-data networking. In: *Proceedings of NDSS Workshop on Security of Emerging Networking Technologies (SENT)* (2014)
12. Grassi, G., Pesavento, D., Pau, G., Vuyyuru, R., Wakikawa, R., Zhang, L.: Vanet via named data networking. In: *Proceedings of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHP)* (2014)
13. Grassi, G., Pesavento, D., Pau, G., Zhang, L., Fdida, S.: Navigo: Interest forwarding by geolocations in vehicular named data networking. In: *Proceedings of International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)* (2015)
14. Hamdane, B., Serhrouchni, A., El Fatmi, S.G.: Access control enforcement in named data networking. In: *Proceedings of International Conference for Internet Technology and Secured Transactions (ICITST)* (2013)
15. Hamdane, B., Serhrouchni, A., Fadlallah, A., Fatmi, S.G.E.: Named-data security scheme for named data networking. In: *Proceedings of International Conference on The Network of the Future (NOF)* (2012)
16. Ion, M., Zhang, J., Schooler, E.M.: Toward content-centric privacy in icn: Attribute-based encryption and routing. In: *Proceedings of ACM SIGCOMM Workshop on Information-centric Networking* (2013)
17. Jin, T., Zhang, X., Liu, Y., Lei, K.: Blockndn: A bitcoin blockchain decentralized system over named data networking. In: *Proceedings of International Conference on Ubiquitous and Future Networks (ICUFN)* (2017)

18. Kim, D., Nam, S., Bi, J., Yeom, I.: Efficient content verification in named data networking. In: Proceedings of ACM Conference on Information-Centric Networking (2015)
19. Larsen Michael, G.F.: Port randomization Work in Progress (2009)
20. Ma, M., Shi, G., Li, F.: Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the Iot scenario. *IEEE Access* **7**, 34045–34059 (2019). <https://doi.org/10.1109/ACCESS.2019.2904042>
21. Mannes, E., Maziero, C.: Naming content on the network layer: A security analysis of the information-centric network model. *ACM Computing Surveys* **52**(3), 44:1–44:28 (2019)
22. Milutinovic, M., He, W., Wu, H., Kanwal, M.: Proof of luck: An efficient blockchain consensus protocol. In: Proceedings of the 1st Workshop on System Software for Trusted Execution, SysTEX '16, pp. 2:1–2:6. ACM, New York (2016). <https://doi.org/10.1145/3007788.3007790>
23. Muzammal, M., Qu, Q., Nasrulin, B.: Renovating blockchain with distributed databases: An open source system. *Futur. Gener. Comput. Syst.* **90**, 105–117 (2019)
24. Alexiou, N., Basagiannis, S., Katsaros, P., Dashpande, T., Smolka, S.A.: Formal analysis of the kaminsky dns cache-poisoning attack using probabilistic model checking. In: 2010 IEEE 12th International Symposium on High Assurance Systems Engineering, pp. 94–103. <https://doi.org/10.1109/HASE.2010.25> (2010)
25. Nguyen, T., Mai, H., Doyen, G., Cогranne, R., Mallouli, W., d. Oca, E.M., Festor, O.: A security monitoring plane for named data networking deployment. *IEEE Communications Magazine* **56**(11), 88–94 (2018)
26. Nurgaliev, I., Muzammal, M., Qu, Q.: Enabling blockchain for efficient spatio-temporal query processing. In: Hacid, H., Cellary, W., Wang, H., Paik, H.Y., Zhou, R. (eds.) *Web Information Systems Engineering – WISE 2018*, pp. 36–51. Springer International Publishing, Cham (2018)
27. Ouaddah Aafaf, A.E.A., Abdellah, A.O.: Fairaccess: a new blockchain-based access control framework for the internet of things. *Security and Communication Networks* **9**(18), 5943–5964 (2016). <https://doi.org/10.1002/sec.1748>. <https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1748>
28. Wang, Q., Chen, D., Zhang, N., Qin, Z., Qin, Z.: Lacs: A lightweight label-based access control scheme in iot-based 5g caching context. *IEEE Access* **5**, 4018–4027 (2017). <https://doi.org/10.1109/ACCESS.2017.2678510>
29. Qu, Q., Nurgaliev, I., Muzammal, M., Jensen, C.S., Fan, J.: On spatio-temporal blockchain query processing. *Futur. Gener. Comput. Syst.* **98**, 208–218 (2019)
30. Saha, B.K., Misra, S.: Mitigating NDN-based fake content dissemination in opportunistic mobile networks. *IEEE Transactions on Mobile Computing* (2019)
31. Saxena, D., Raychoudhury, V., Becker, C.: Implementation and performance evaluation of name-based forwarding schemes in v-NDN. In: Proceedings of International Conference on Distributed Computing and Networking (2017)
32. Shang, W., Ding, Q., Marianantoni, A., Burke, J., Zhang, L.: Securing building management systems using named data networking. *IEEE Netw.* **28**(3), 50–56 (2014)
33. Singh, V.P., Ujjwal, R.L.: Privacy attack modeling and risk assessment method for name data networking. In: Bhatia, S.K., Tiwari, S., Mishra, K.K., Trivedi, M.C. (eds.) *Advances in Computer Communication and Computational Sciences*, pp. 109–119 (2019)
34. Wang, L., Afanasyev, A., Kuntz, R., Vuyyuru, R., Wakikawa, R., Zhang, L.: Rapid traffic information dissemination using named data. In: Proceedings of ACM Workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications (2012)
35. Wang, S., Xing, Z., Yan, Z., Lin, W., Wang, W.: A survey on mobile edge networks: Convergence of computing, caching and communications. *IEEE Access* **5**(99), 6757–6779 (2017)
36. Wright, A., De Filippi, P.: Decentralized blockchain technology and the rise of lex cryptographia. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.2580664> (2015)
37. Yang, H., Cha, H., Song, Y.: Secure identifier management based on blockchain technology in NDN environment. *IEEE Access* **7**, 6262–6268 (2019)
38. Yi, C., Afanasyev, A., Wang, L., Zhang, B., Zhang, L.: Adaptive forwarding in named data networking. *ACM SIGCOMM Computer Communication Review* **42**(3), 62–67 (2012)
39. Yin, H., Zhang, J., Xiong, Y., Ou, L., Li, F., Liao, S., Li, K.: Cp-abse: A ciphertext-policy attribute-based searchable encryption scheme. *IEEE Access* **7**, 5682–5694 (2019)
40. Yu, Y.: Public key management in named data networking. *Tech. Rep NDN-0029* (2015)
41. Yu, Y., Afanasyev, A., Zhu, Z., Zhang, L.: An endorsement-based key management system for decentralized NDN chat application. *Tech. Rep NDN-0023* (2014)
42. Yu, Y., Dilmaghani, R.B., Calo, S., Sanadidi, M.Y., Gerla, M.: Interest propagation in named data manets. In: Proceedings of International Conference on Computing, Networking and Communications (ICNC) (2013)
43. Yu, Y., Gerla, M.: Potential benefits of information-centric networks for vanets (2015)
44. Zhang, Z., Yu, Y., Ramani, S.K., Afanasyev, A., Zhang, L.: Nac: Automating access control via named data. In: Proceedings of IEEE Military Communications Conference (MILCOM) (2018)
45. Zhu, Z., Wang, S., Yang, X., Jacobson, V., Zhang, L.: Act: Audio conference tool over named data networking. In: Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking (2011)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.