IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 22, 2020, accepted April 28, 2020, date of publication May 4, 2020, date of current version May 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992115

Smart Contract-Based Trusted Content Retrieval Mechanism for NDN

TINGTING SONG[1,2], BO CUI[1,2], (Member, IEEE), RU LI[1,2], (Member, IEEE), JING LIU[2], (Member, IEEE), AND JINSHAN SHI[1,2]

[1]Inner Mongolia Key Laboratory of Wireless Networking and Mobile Computing, Hohhot 010021, China
[2]College of Computer Science, Inner Mongolia University, Hohhot 010021, China

Corresponding author: Bo Cui (cscb@imu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61962042 and Grant 61862046, in part by the Natural Science Foundation of Inner Mongolia under Grant 2018MS06028 and Grant 2019MS06007, in part by the China Education and Research Network (CERNET) Innovation Project under Grant NGII20170415, and in part by the Science and Technology Program of Inner Mongolia Autonomous Region under Grant 2019GG376.

ABSTRACT Named Data Networking (NDN) is a new clean-slate architecture for the future Internet. Efficient content retrieval is the original intention of NDN design. The content retrieval process driven by content consumers in NDN includes the following challenges, consumers do not know whether the content exists and whether the content producer is reliable. Invalid interest packets could cause the occupation of limited network resources and DoS attack problem. To ensure the authenticity and integrity of the data packets, consumers need to pre-configure the trust schema, which is centralized and prone to the single point of failure problem. Blockchain has widespread attention to build trust in a distributed way, and Ethereum is a programmable blockchain, a decentralized smart contract platform. To lighten the burden of consumers, we proposed a Smart Contract-based Trusted Content Retrieval Mechanism (SCTCRM) for NDN in this paper. The mechanism contains a trustworthy information base for content and producers based on smart contracts, and provides content retrieval and name resolution services for content consumers. The purpose of this mechanism is to improve the efficiency and security of content retrieval process. We described the framework and the workflow of SCTCRM, and used Colored Petri Nets to create a formal mathematical model and analyze the security of the mechanism. Finally, the cost of storage and Gas in smart contracts are evaluated through the prototype deployment. From the results, we can see that the proposed mechanism is security and practicality.

INDEX TERMS Named data networking, blockchain technology, smart contract, trusted content retrieval, Colored Petri Nets.

I. INTRODUCTION

Named Data Networking (NDN) [1] is a clean-slate design for the future Internet, which inherits the hourglass model of the TCP/IP architecture, but the narrow waist leverages names of data chunks instead of IP addresses for data delivery. In the TCP/IP network architecture, communication pattern is end to end, the user could find the destination host through the domain directory provided by DNS [2]. NDN focuses on retrieving interested content rather than finding a specific and physical location where the content is.

Efficient content retrieval is the original intention of NDN design. The content retrieval process is driven by the content consumer in NDN. The process has two challenges. One is that consumers do not know whether the data they interested exists, which could cause the invalid interests occupy the limited space of routers and DoS attack problem [3]–[5]. The other is that consumers do not know whether the content producer is reliable, to verify the identity of content producer, they need to pre-configure the trust schema [6].

There have been some researches in content name retrieval for NDN, a DNS-like lookup service for NDN through sending specific packets was proposed in [7], too many invalid request packets could also cause DoS problem. In [8], blockchain technology is introduced to design a hybrid chain based hierarchical name resolution service for NDN, which designed a new type of namespace combined with the block height and the transaction number, which is unreadable. However, the content name in NDN is designed to be readable

The associate editor coordinating the review of this manuscript and approving it for publication was Srinivas Sampalli.

and meaningful in the contexts. So how to design an effective content retrieval mechanism combined with the naming characteristics of NDN is need to be solved.

In the terms of the trust schema, Yu *et al.* [6] introduced the trust management schema to manage the trust relationships of the content and producers. However, the centralized architecture of trust schema proposed in this paper is prone to the single point of failure problem. Inspired by the concept of Web-of-Trust (WoT), a distributed trust schema is proposed in NDN [9], this schema has certain restriction conditions for application. Blockchain technology is a shared, unalterable ledger for recording transactions, tracking assets and building trust in a distributed approach [10], and brings a new solution to resolve the single point failure problem. It has drawn widely concern in the field of digital certification and authentication [11]–[16], and has been introduced into NDN to improve the security and efficiency of network. In [17], blockchain as a trustworthy database for keys management in NDN, the way to obtain the data stored in the blockchain is retrieving the block number and the hash value of the transaction, which will affect the efficiency of retrieval when the number of transactions is large. Thus, how to improve the efficiency of content retrieval is another problem that needs to be solved.

Ethereum is a programmable blockchain platform [18], which makes the blockchain could not only act as a distributed ledger, but also support users to deploy the functional smart contracts and distributed applications. The Ethereum's smart contracts could be written in the high-level programming language Solidity [19], solidity events provide of the logging facilities interfaces of Ethereum virtual machine (EVM), which could be listened effectively through the RPC interface of the Ethereum client.

A Smart Contract-based Trusted Content Retrieval Mechanism (SCTCRM) for NDN is proposed in this paper by using blockchain technology and Ethereum smart contracts. The primary purpose of the mechanism is to build a distributed global content and producer information storage structure and a name searching service to help content consumers find out the corresponding content and the reliable content producer in NDN. The mechanism utilization the tempering resist feature of blockchain to ensure the authenticity and integrity of data and build trust in a decentralized way, and for privacy, the encryption protocol is introduced to protect the confidentiality of data. By using the smart contracts, the information stored in the blockchain can be managed, and the user's operations to the smart contracts are traceable. The named smart contract searching algorithm designed in this mechanism is to improve the efficiency of content retrieval. The proposed mechanism could help to reduce the possibility of obtaining fake data packets from attackers by finding reliable content producers.

The contributions of this paper are as follows:

1. We introduced the framework design and the main work flow of SCTCRM, and built the formal model of the mechanism and analyzed the security of the model.

2. The smart contract prototype of SCTCRM is deployed on the consortium blockchain, and the cost of storage and the Gas in smart contracts were tested through the implement experiments.

The rest of this paper is organized as follows: Some necessary background notions about NDN, blockchain technology and Ethereum are introduced in Section II. The problem definition is introduced in Section III. The network architecture, the main work flow and the named smart contract searching algorithm of SCTCRM are shown in Section IV. The formal model build by Colored Petri Nets and the security analysis of SCTCRM are presented in Section V. The implement of the prototype of the mechanism and the overhead evaluation are discussed in Section VI. The related work is discussed in Section VII. Finally, the conclusions and future work are presented in Section VIII.

II. BACKGROUND

In this section, first we briefly review the NDN architecture, then we provide an introduction of blockchain technology and some preliminary concepts of Ethereum smart contracts in order to give a precise description of the proposed mechanism in this paper.

A. NAMED DATA NETWORKING

NDN is an alternative approach to the TCP/IP based Internet architecture. The communication pattern is fundamentally different from the existing network. NDN has attracted increasing attention in many field recent years, with the potential to outperform the current IP paradigm [20], [21].

In NDN, the user only needs to focus on the content of interest, rather than the location where the content can be found. The data is named for content search and retrieval. There are only two kinds of packets, e.g. Interest packet and Data packet, and three main types of entities including data consumer, producer and router/forwarder. Moreover, each NDN node maintains three major data structures: Pending Interest Table (PIT), Forwarding Information Base (FIB), and Content Store (CS) [1]. Communication is driven by the consumer, which uses an Interest packet to request content by a name. Any node who has the matching content replies with a Data packet that contains the named content and additional signature information.

B. BLOCKCHAIN AND ETHEREUM

Blockchain technology is a distributed ledger technology from bitcoin [22], which is a combination of a series of technologies, including distributed data storage, consensus mechanism, cryptography, P2P network transmission, etc. Blockchain has the linked list structure, each block stores transaction data for a period of time, these blocks are linked together in chronological order to form a linked list. The transaction data of blockchain is open and publicly shared to the blockchain users, and the cryptography and consensus mechanism are used to ensure that the transaction data cannot be tampered with. The characteristics of blockchain make

it possible to build trust between nodes in a decentralized system without a trusted third party, which provides a solution to the single point of failure of the traditional centralized trust structure.

As a programmable blockchain, Ethereum allows the users to create, deploy and run smart contracts or decentralized applications on the blockchain [18], [23]. The smart contract is written in a high-level language such as solidity, and compiled into bytecode, which is executed by Ethereum virtual machine (EVM), the core part of Ethereum. The calculation in EVM is paid by ether (ETH) using Gas as its unit. The deployment of smart contracts and the calling the functions have Gas cost, which depends on the calculation complexity and the storage space. The events protocol of smart contracts provides an abstraction of the EVM's logs, which could be subscribed and listened through the RPC interface by the Ethereum clients. Externally Owned Account (EOA) and Contract Account are two types of accounts in Ethereum. The EOA has address, a pair of public key and private key, and ether balance, which is created by people and could send transactions. The Contract Account represents the smart contract.

III. PROBLEM DEFINITION AND PRORERTIES OF SCTCRM

In this section, we discuss the requirements of content consumers during the content requesting process and the security requirements during the content receiving process. Then we describe the properties of SCTCRM in order to achieve those requirements.

A. PROBLEM DEFINITION

1) RETRIEVAL ISSUES DURING THE CONTENT REQUESTING PROCESS

The content retrieve process in NDN is driven by the interest packets, routing and forwarding protocols. When there is no content corresponding to the name of interests, those invalid interests will occupy the limited PIT and FIB of the routers. If consumers could directly search whether the content they need is exists, which will help to reduce the burden of invalid interests. Moreover, if consumers could find a reliable content producer, the risk of receiving malicious data packets from attackers will also be reduced. We should consider the appropriate content search method for consumers to facilitate the content retrieve efficiency.

2) SECURITY ISSUES DURING THE CONTENT RECEIVING PROCESS

NDN is a data-centric security network architecture. During the data packets forwarding progress, there are mainly the following aspects of security analysis.

The integrity of content: When the data packets entered into the forwarding process without the supervision of producers, they could be tampered or replaced by attackers. Therefore, a security protocol should be proposed to protect the integrity of the valid data packets sent by reliable producers.

The authenticity of content: It is necessary to verify the authenticity of content producers, which determines the security of data and also, we should ensure the relevance of the name and the content.

Content non-repudiation: The content producer should not deny the data packets they send. Thus, a monitoring protocol should be established to avoid the false content injection attack.

The confidentiality of content: For the data packets that need to be kept secret, only the authorized consumers could obtain the content from them. The encrypted protocol should be used to protect the confidentiality of private data packets.

B. PROPERTIES OF SCTCRM

In this paper, the smart contract-based trusted content retrieval mechanism mainly focuses on retrieve the information of content and producers. In order to solve the problems mentioned above, the proposed mechanism should have the following properties.

Efficiency: The smart contracts are used to store the description information of content and the identity information of content producers. The traditional method to traverse blockchain grounded on block number combined with transaction number has high time complexity, Thus, we take the naming feature in NDN into consideration, and create a name prefix search tree to improve the searching efficiency by matching the content name prefix and the name of smart contracts.

Integrity: We choose to store the hash value of the data packets, i.e. content digest, in the blockchain to protect the content not be tampered with. For the public and static content, producers could publish the content digest to the smart contracts directly. And consumers could compare the hash value of the received data packets with the content digest stored in the blockchain to verify the integrity of the content. For the dynamically generate content, consumers could send transactions to request content digest from producers, and producers should response before they send the data packets. For the private data, producers should use the public key of authorized consumers to encrypt the content digest. The content digest obtained in advance is to help consumers to confirm the integrity of data packets.

Authenticity: After the consumer retrieval the content name, SCTCRM will match the name of the smart contracts according to the named prefix tree, and obtain the addresses of smart contracts. By accessing to the smart contracts through the addresses, the content information and the identity information of producers could be fetched.

Non-repudiation: We use the events and log protocol of smart contract to monitor every action triggered by the owner of the smart contract. Those log records can be used to trace the operations performed by the content producer.

Confidentiality: For the private content, the privacy-related information should not be stored in the smart contracts. Consumers could request content producers for privacy information, such as content digest, through the transactions of

TABLE 1. Functional description of main roles.

Roles	Functional description
Ethereum nodes	Store and execute smart contracts to provide users with search services.
Content Producers	Provide content, and deploy a smart contract corresponding to the content or their identity on the Ethereum blockchain.
Content Consumers	Send interest packets, and search the content information or the producer identity information from blockchain through the name prefix.
Route nodes	Exchange and cache the interests and content.

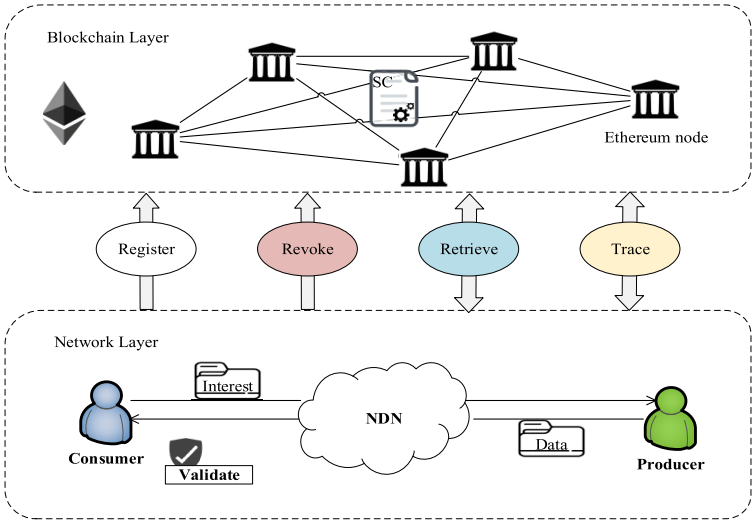


FIGURE 1. The network architecture of SCTCRM.

Ethereum. The producer should verify the identity of the consumer, and use the public key of the consumer to encrypt the privacy information.

IV. THE SMART CONTRACT-BASED TRUSTED CONTENT RETRIEVAL MECHANISM FOR NDN

In this section, we introduce the framework of SCTCRM, and describe the main work flow and the smart contracts design of the mechanism. We also propose the algorithm design of improve the content searching efficiency, and analyze the time complexity.

A. THE FRAMEWORK OF SCTCRM

The scenario of SCTCRM is to provide a trustworthy content retrieval service for NDN users. The naming structure of content is readable and hierarchy, a content could be uniquely identified by the name prefix of the producer combined with the name of content itself, for example: /IMU/CS/Software/source. If the relationship of name, content, and producers could be publicly known, consumers could find the trustworthy data and reliable producers independently by searching the name of content, which is helpful to lower the possibility of obtaining the malicious data packets.

There are mainly four types of roles in the scenario, which are Ethereum nodes, content producers, content consumers and routers. The functional description of the main roles is shown in Table 1. Ethereum nodes contain three major functional modules: blockchain database, EVM, and the content retrieval module. The blockchain database is used to store the complete data of blockchain, EVM is used to compile and execute smart contracts, and the content retrieval module is designed for searching whether there is content information corresponding to the content name. Content consumers represent consumers in NDN, who request for data by sending interest packets, and could send content retrieval request in our scenario. Content producers represent producers in NDN, who produce and provide data by sending data packets. And in our scenario, they also could deploy the smart contracts on the Ethereum blockchain which publish their identity and the content information they could provide. Route nodes represent the routers in NDN, which are mainly responsible for exchange and cache the interest packets and data packets.

The network architecture of SCTCRM is shown in Figure 1. The network layer is mainly about exchanging the interest packets and the data packets in NDN. The application layer is based on Ethereum platform and smart contracts to implement the user's registration, retrieval, revocation and event log tracing functions.

Smart contracts store the content-related information, the identity information of content producers, and the mapping relationship between them. The name of the smart contract is exactly corresponding to the content name. After producers deployed the smart contracts and obtained the consensus of

the Ethereum nodes, the smart contracts will be stored as transactions in the blockchain. The blockchain technology ensures the security and credibility of the information in this scenario. The smart contracts provide effective, efficient and secure implementation of different functions which executed exactly as the program.

After the smart contracts are deployed and stored in the blockchain, the content name retrieve module will build a name prefix tree for all of the smart contract names, which mapping the names and the addresses of smart contracts.

B. THE WORK FLOW AND SMART CONTRACTS DESIGN OF SCTCRM

There are two types of smart contracts in this paper, the identify certification smart contract stores the identity information of content producers and the content smart contract stores the content publicly information.

User's identity certificate in NDN is implemented in the form of a data packet. The certificate content data packet carried the name of the producer, the public key of the producer as the content of it, and the signature which signed by the higher-level user. In SCTCRM, a smart contract could be able to represent an identity certificate, because of the smart contract contains name, data and the smart contract deployment transaction requires the signature signed by publisher. The signature could represent the permission license of user's identity. The smart contracts could also provide the revocation of user's identity and the event listener logs of user's operations.

Since the hierarchical naming structure of the NDN application could be regarded as a tree structure, in the process of distributing identity certificates, users could be divided into a root user and the leaf users. The root user could publish a self-signed identity certificate smart contract, and the other leaf users need to apply for the identity certificate registration from a higher-level user. The higher-level user could use *challenge-response* protocol to verify the eligibility of the applicant through the Ethereum transactions. If the verification is succeeded, the higher-level user will publish the identity certificate smart contract of the applying user on the blockchain. The identity certificate smart contract registration process is shown in Figure 2.

The smart contract is a combination of code and data, which should be designed carefully. The identity certificate smart contract could store the following identity information <name prefix, producer's name, producer's NDN public key, producer's Ethereum address, time stamp, validity period, auth's Ethereum address, status>. The name prefix and the producer's name could uniquely identify the user. The producer's NDN public key could verify the signature of the data packet. The producer's Ethereum address could be used to support the *challenge-response* protocol. The time stamp and validity period indicate the duration of the identity certificate. The auth's Ethereum address indicates the publisher of the smart contract. The status indicates that whether the contract is valid, that is, whether the producer's identity has been revoked. The reason for choosing to generate a smart contract for each user's identity to represent an identity certificate is that every user's identity certificate is corresponding to a unique smart contract transaction with signature, and this is convenient for building the name prefix search tree.

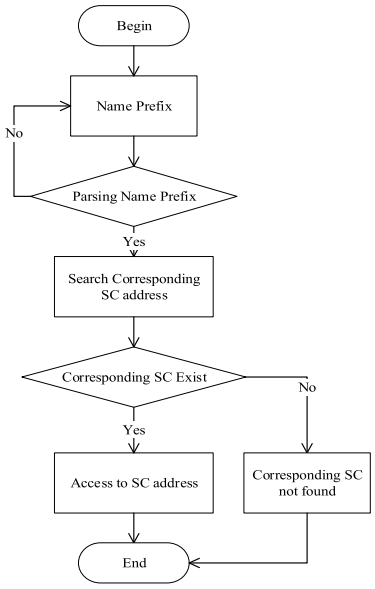


FIGURE 2. Identity certificate smart contract registration process.

The functions of identity certificate smart contract include apply for identity registration, revoke producer's identity and return the identity information stored in the blockchain. The identity registration function is for other users to provide identity information when they apply for a new identity certificate from the higher-level user. Considering that users might reveal their NDN private key or have malicious behavior, the users' identity could be revoked by the higher-level users who publish the identity certificate smart contract.

The user who successfully registers the identity certificate smart contract is defined as a reliable content producer, who could generate and deploy the content contract containing the details and security information of the content. SCTCRM also generate a smart contract for each content, mainly due to the following factors. Firstly, a large amount of content could be generated and revoked at the same time, which will cause a huge burden of update and maintenance overhead for creating a single smart contract for several content. Secondly, it is more secure for each content smart contract mapping with the producer, ensuring the feature of content non-repudiation.

The content smart contract could store the following content information <content prefix, content name, content digest, producer's Ethereum address, producer's identity certificate address, time stamp, validity period, status>. The content prefix and content name uniquely identify the content. The content digest is the hash value of the data packet, which is used to verify the content integrity.

If the content is static and publicly available, the producer could publish the content digest directly, otherwise the

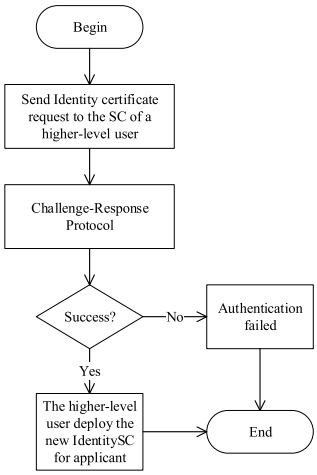


FIGURE 3. Name retrieval process.

producer should not publish the content digest. The producer's Ethereum address could support consumers send the content digest request transaction to the producer when the digest is unpublished. The producer's identity certificate address indicates that the mapping relationship between the producer and the content, which ensures the security of the content. The time stamp is the publish time of smart contract, the validity period indicates the available period of the content, and the status shows whether the content is revoked. The functions of content smart contract include add content information, revoke content and return content information.

The blockchain node constructs a named prefix search tree of the contract name, contract ABI and contract address of all kinds of smart contracts. The named retrieval process is shown in Figure 3.

Different name prefixes correspond to different types of smart contracts. For example: the name of /IMU/CS/Software indicates an identity certificate smart contract, the name of /IMU/CS/Software/source indicates a content smart contract.

Consumers need to request the required name and find out the Ethereum address, then figure out more valuable information. The content information acquisition process is shown in Figure 4.

The authenticity and integrity of the data packets are verified by the immutable storage of the data in the blockchain, the identity information of content producers, the content digest information, and the mapping relationship between them. The interest packets could also add a content digest part which recorded by the routers, and choose the self-verify schema [24] to ensure the integrity of the data packets, enhance the security of network layer.

C. THE NAMED SMART CONTRACT SEARCHING ALGORITHM

Blockchain is a single linked list data structure, which consists of a series of connected data blocks. Each block consists of two parts, block head and block body. Block header encapsulates hash value of previous block, time stamp, nonce, root value of Merkle tree, hash value of current block and some

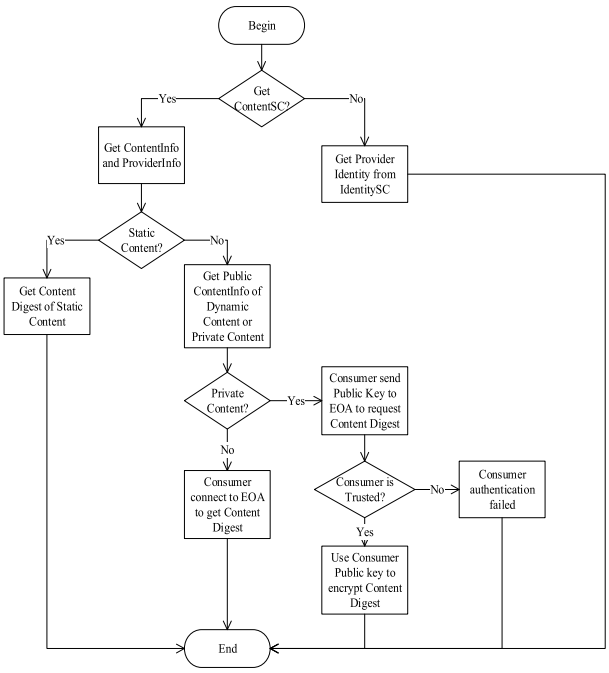


FIGURE 4. The content digest request process.

other information. Block body mainly contains transaction count and transaction details. Among them, the hash value of the previous block as a pointer, connecting the blocks from the genesis block to the latest block.

The time complexity of searching for data in the single linked list data structure is usually about linear time O(n), where n is the number of blocks. However, in fact, in the process of obtaining data, we need to fetch the information stored in the transaction, each block contains multiple transactions. Since the transaction is stored in the Merkle tree, the time to query data on the blockchain becomes O(nlogm), where n is the number of blocks and m is the transaction number in every block. The query efficiency of traditional algorithm for querying information in blockchain is low, and the algorithm is described as Algorithm 1.

The searching efficiency of the traditional transaction searching algorithm is not ideal. Therefore, a named smart contract searching algorithm was proposed in this paper. The name prefix tree in the algorithm is to improve the searching efficiency of the smart contracts and combine the naming structure feature in NDN.

Blockchain in the algorithm is act as a data source of the content names, which builds the mapping relationship of all of the smart contracts names and addresses to create a name prefix tree. The purpose is to search for the smart contract address corresponding to the name of content or producers. Therefore, consumers could find out whether the content they interested or the reliable producers exist in the network. After accessed the smart contracts through the addresses, consumers could obtain information stored in the blockchain.

Figure 5 shows an example of name searching process and the architecture of name searching module.

Algorithm 1 Traditional Blockchain Transaction Searching Algorithm

Input:
The transaction t hash value h
Output:
The transaction t details *transInfo*
flag = 0 //mark whether the transaction is found
transInfo = null //store the details of transaction t
reqhash = h
i = CurrentBlockNum
// Synchronize the blockchain with the last block number of blockchain(i = 1,2,. . . ,n)
while flag = 0 && i > 0 **do** / / traverse all of the blocks
  j = 0
  for each $Block_i$ **do**
    if reqhash = $Block_i.transNum_j$ **then**
      // If the hash value of the transaction is found
      flag = 1
      transInfo = $Block_i.transInfo_j$
      break
    end if
    j = j + 1
  end for
end while
if flag = 0 **then**
  return error // The transaction t does not exist
else
  return transInfo
    // return the details of the transaction t
end if

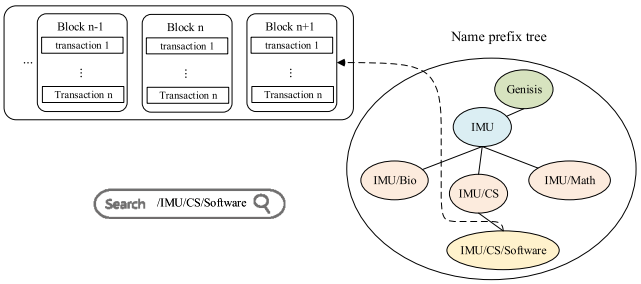


FIGURE 5. An example of name searching module.

The user searched the name of /IMU/CS/Software. Each node of the name prefix tree represents the mapping of content name and its smart contract. Finally, the target smart contract is found after searching and matching the correct node of tree. The proposed algorithm is as follows Algorithm 2.

The name prefix tree is similar to the dictionary tree Trie. If assuming that the name prefix tree has k levels, the time complexity of searching the smart contract is O(k).

V. SCTCRM FORMAL MODEL AND SECURITY ANALYSIS

In this section, we use the Colored Petri Nets (CPN) tool to formalize the above SCTCRM, through hierarchical CPN in consider of the complexity of the mechanism. The relevant concepts of CPN can be referred to the literature [25].

Algorithm 2 Named Smart Contract Searching Algorithm

Input:
BlockChain (t), Name //time t,content or producer name
Output:
ContentInfo or ProducerInfo
NameTree (t) = TreeInitialise (BlockChain (t))
// Build a name tree based on the blockchain at time t
while blockchain consensus
// Update the name tree periodically
  if a new contract is deployed **then**
    NameTree (t + 1) = AddNode (BlockChain (t + 1))
  else if there is an contract to be destroyed **then**
    NameTree (t + 1) = DeleteNode (BlockChain (t + 1))
  end if
end while
for genesis node to leaf node // Content search
  for the first child node to the end of the content name
    if the node name is equal to the content_name **then**
      access to smart contract from address and call the function
      return content information
      break
    else
      return does not have this content
    end if
  end for
end for

TABLE 2. The color sets of formal model.

Variable	Color set description
colset *NO* = int;	Different kind of token
colset *DATA* = string;	The specific name of token
colset *NOxDATA*= product *NO* **DATA*;	Full form of token transmitted

There are three types of color sets according to the formal model of SCTCRM. The definition and description of color sets are explained in Table 2.

We analyze the work flow of SCTCRM and summarize the attributes of the model as follows:

1. The content consumer could send content name search request all the time.
2. Only the user who pass the challenge-response authentication protocol could have identity certificate.
3. The content producer cannot deploy the content smart contracts without obtaining identity certificates.
4. After the smart contract is deployed, the information of blockchain should be updated.
5. After the interest packet is send to the network, the corresponding data packet could be returned.

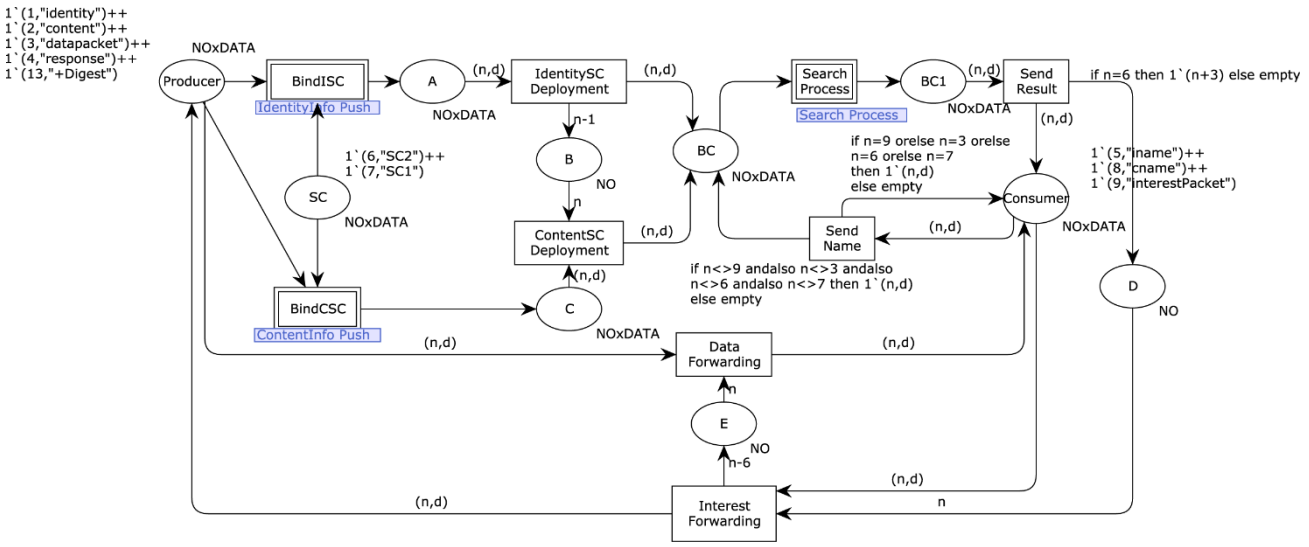


FIGURE 6. The top layer CPN model of SCTCRM.

The top layer model is shown in the Figure 6. There are 9 transitions and 10 places in the model. The place *Producer* represents the content producer, the place *Consumer* represents the content consumer, the place *BC* represents the blockchain, and the place *SC* represents the smart contracts. The transition *BindISC*, *BindCSC and Search Process* are substitution transitions.

BindISC indicates the process that producer apply for the registration of the identity certificate. If the producer could pass authentication of the higher-level user, the *IdentitySC Deployment* will be fired, which means the identify certificate smart contract will be deploy on the blockchain. *BindCSC* indicates that the producer publishes the content information, and the transition *ContentSC Deployment* is fired to deploy the content smart contract on the blockchain. The place *B* is to limit that only the registered producers can deploy the content smart contracts. The place *A* and *C* is used to transfer the contract. The transition *Send Name* represents consumers send the name retrieval request about the about content or producers they interested. The name will be transfer into *Search Process*, which indicates the process of retrieving the name prefix tree to match the corresponding smart contract. If the smart contract could be found, the information requested will be return to the consumer after the *Send Result* fired. The place *BC1* represents the information of smart contract is returned. If the consumer gets the content information corresponding to the name they searched, they could send an interest packet carry the name to request for data, the transition *Interest Forwarding* and *Data Forwarding* will be fired to exchange the interest packet and the data packet. After receiving the interest packets, content producers send data packets to fire Data Forwarding transition which will be forward back to the consumer. The place D is to transfer the interest packet. The place E is to limit that the data packets will be returned only after the producer receive the interest packet.

The *BindISC* layer is the bottom implementation of the *BindISC* substitution transitions, which is modeled for the content producer apply for the registration of the identity certificate shown in Figure 7. There are 5 transitions and 8 places in the model, where the place *Producer*, *SC*, *A* are the interfaces connected to the top layer. The content producer chooses the publicly available content information which could be stored in the blockchain through firing the transition *Get ContentInfo*. The transition *Get2* is to transfer the data packet. If the content is static and publicly available, the producer could publish the content digest which is the hash value the data packet created by firing the *Hash Packet* transition. The place *contentInfo* is the selected content information. The place *digest* is the content digest. The place *contentfullInfo* is all of the content information that need to be published.

The content producer chooses the publicly available content information which could be stored in the blockchain through firing the transition *Get ContentInfo*. The transition *Get2* is to transfer the data packet. If the content is static and publicly available, the producer could publish the content digest which is the hash value the data packet created by firing the *Hash Packet* transition. The place *contentInfo* is the selected content information. The place *digest* is the content digest. The place *contentfullInfo* is all of the content information that need to be published.

The *Search Process* layer is the bottom implementation of the *Search Process* substitution transitions, which is modeled for the content producer publish the content information into the content smart contract shown in Figure 9. There are 11 transitions and 16 places in the model, where the place *BC*, *BC1* are the interfaces connected to the top layer. When the blockchain receives the request of the content name or producer's name prefix, the transition *Search Name* will be fired to enter the searching process. By transferring to the latest named prefix search tree, and the transition *Prefix Query* is fired to match the name requested and the name of smart

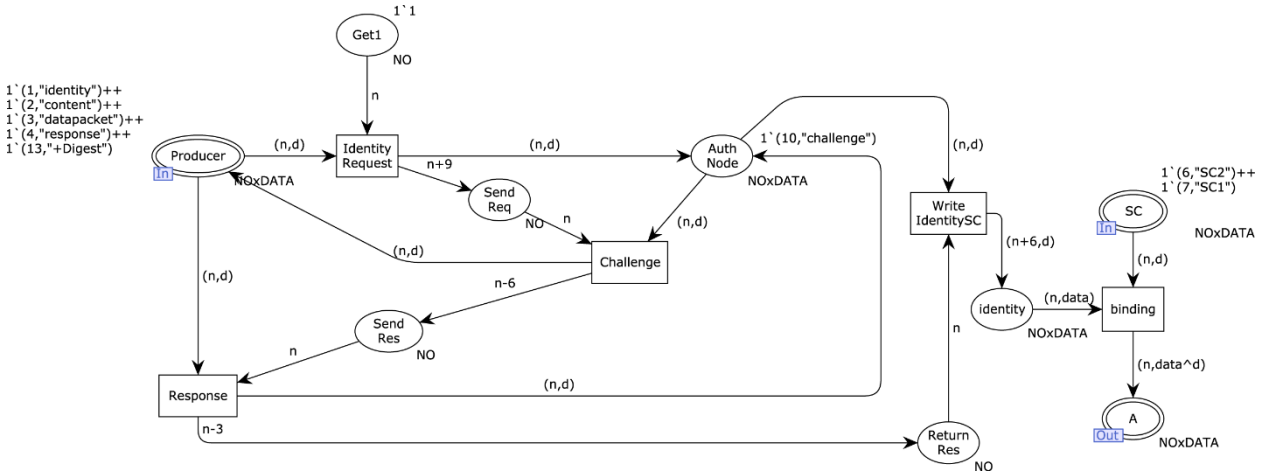


FIGURE 7. The BindISC layer.

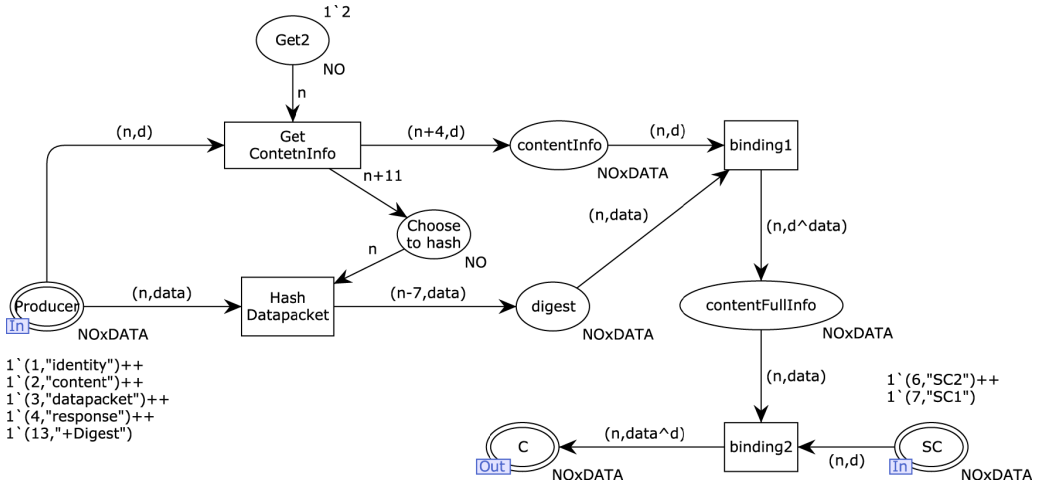


FIGURE 8. The BindCSC layer.

contracts. After obtaining the address of the corresponding smart contract, the transition r*eturnIDSC* or *returnCSC* will be fired to access to the smart contract by address and return the request information.

To simulate and analyze state space of the model, the simulation proves that the processes of the mechanism are expected as the attributes we summarized before. There is no dead state or dead transition in the model, and there is only a live transition Send Name, that is, under all of the reachable transition of the model, the transition *Send Name* is in the sequence of occurrence. No matter which state the model is in, the user could send the name request to the blockchain, which conforms to the actual situation. In the state space, the number of tokens allowed by each place is analyzed, the bounded of the amount tokens is in accordance with the requirements of the model.

VI. IMPLIMENTATION AND ANALYSIS RESULTS

In this section, we represent the deployment of the prototype of SCTCRM, and then we discuss the overhead of introducing the smart contract. From the results we can see that the proposed mechanism is practicality.

A. IMPLIMENTATION DETAILS

In this paper, the content producer and the content consumer correspond to External Owned Account (EOA) of Ethereum. Since the content users involved in this mechanism include root-level users of NDN applications and other level users, the consortium blockchain is more appropriate in the selection of Ethereum. A large number of block data need to be synchronized in the public blockchain will waste amount of storage usage of NDN user's devices, and the private blockchain does not have enough decentralization compared to the public blockchain and the consortium blockchain. In SCTCRM, the root-level users of each application are corresponding to the nodes of the consortium blockchain, the other level users correspond to the external accounts of each consortium blockchain node, and the consortium blockchain is managed by the root-level users of each application.

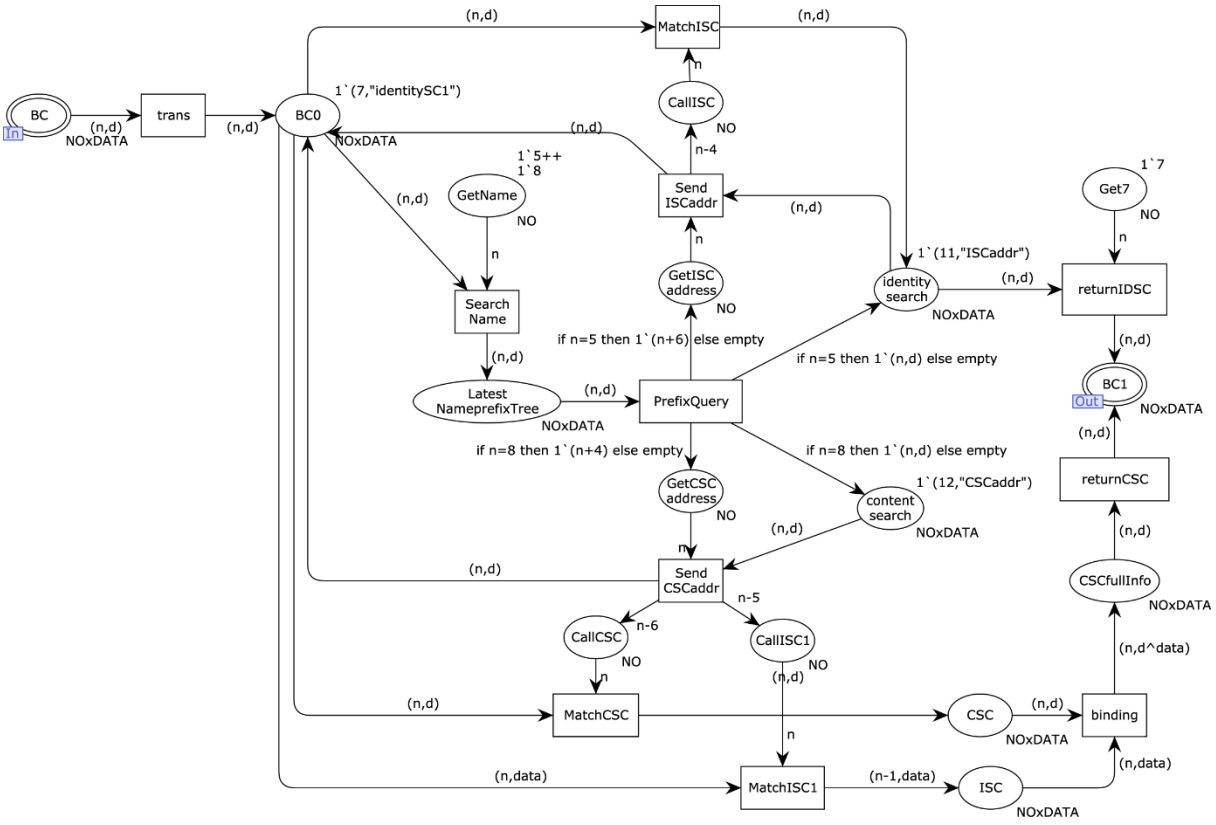


FIGURE 9. The search process layer.

The smart contracts are implemented in Solidity, a high-level programming language for writing Ethereum smart contracts that are compiled to EVM code. We conduct the experiments via the *truffle* suite [26], a testing framework that automatically handles compilation and deployment of contracts. We choose *geth* [27] as our Ethereum consortium nodes running on the machine. The *MetaMask* [28] browser plug-in Ethereum wallet is used to support the client of SCTCRM, web3.js is used to interact with the smart contracts, and node.js is used to complete the information interaction between the client and the consortium blockchain nodes. In the deployment process, the name search module is built locally to improve retrieval efficiency, which stores the name, ABI and address of the smart contract under the local name folder and updating the local information synchronously with the blockchain. The name search module retrieves the matching local corresponding name file path by naming prefix, and accesses and calls the corresponding smart contract by using the retrieved smart contract information.

In Ethereum, computing and storage are relatively expensive, so experiments are needed to test the cost of smart contracts and storage data. The Solidity language is used to create smart contract to register the identity of the content producer and content information. Smart contracts are compiled in EVM bytecode and deployed in Ethereum. Ethereum uses Merkle Patricia Trie to store the status and data of smart contracts, which can be used to represent the cost of storage.

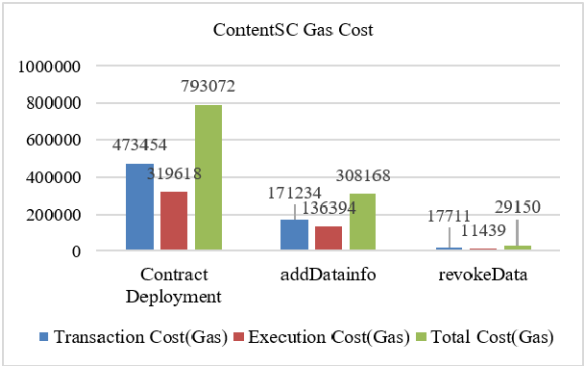


FIGURE 10. The gas cost in content smart contract.

B. ANALYSIS AND RESULTS

We mainly consider about the cost of storage and Gas in smart contracts. The content smart contract takes up about 11,137 bytes of storage space and the identity certificate smart contract takes up about 18,644 bytes of storage space.

The deployment experiment will test the Gas cost in smart contracts. The Gas cost includes the Gas required to create the transaction and to execute the transaction. Figure 10 shows the cost of Gas to create content smart contracts and to call the functions, and Figure 11 shows the cost of Gas for user to register the identity certificate contracts and to call the functions.

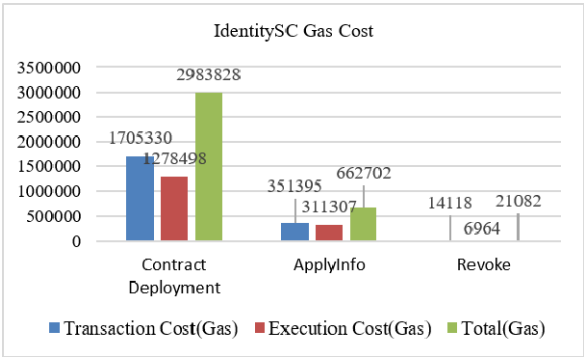


FIGURE 11. The gas cost in identity certificate smart contract.

On March 2020, 1 ether ≈ $242.30, and 1 gas ≈ 1 wei (0.000000001 eth) were used. The deployment of the content smart contract will cost approximately $0.19 to execute the addDatainfo() costs about $0.07, the execution of the revokeData() costs about $0.007, the creation and deployment of the identity certificate smart contract costs about $0.72, the execution of applyinfo() costs about $0.16, and the execution of the revoke() costs about $0.005.

During the release transaction process, Ethereum provides an optional range of ranges for the Gas, the Gas value in verse with the miners' time to verify the transaction, and Gas's cost is proportional to the amount of data uploaded to the blockchain. In SCTCRM, content consumers are expected to obtain query results in a relatively short period of time. The ideal effect is to reduce the execution time of transactions when Gas cost is as small as possible, so we will use the distributed storage platform Swarm or IPFS combined with Ethereum to reduce the amount of data uploaded to the blockchain, so that more Gas could be used to improve the verification transaction time, and also improve the efficiency of interaction with blockchain data.

VII. RELATED WORK

On the issue of content retrieval, the concept of DNS is introduced. The DNS-Like lookup service NDNS pointed out the similarities and differences between the name resolution in NDN and the address resolution in IP network, and also indicated that the multiple independent trust anchors on the top should be supported [7]. The request and response in NDNS are based on packets as the units, consider of the attackers could send a large number of invalid request packets could cause the network requests occupied, which is lead to the DoS problem similar to DNS.

Yu *et al.* [8] use blockchain technology to build a distributed DNS for the peer-to-peer information center network. The ODIN namespace which combined with the block number and the transaction number is created to identify the producer. The consumer use ODIN namespace as the prefix of keywords to search for a content, ppk:351474.430/21.35/ISBN2890321345/1.0. The ODIN namespace is unreadable, unlike the traditional readable naming structure in NDN.

On the issues of identity authentication of content producers, the centralized trust management schema [6] consists of trust anchors and a set of trust rules. The chain of trust is composed of all of the users from the trust anchor to the bottom content producer, if the chain of trust could be verified, the producer is proved to be a reliable user. Before the authentication, trust anchors need to be configured and trust rules need to be learned. And in fact, different applications are allowed to have different trust schemas, which makes it difficult for consumers to verify the credibility of content producers. The centralized architecture of trust schema is also prone to the single point of failure problem.

A distributed trust model based on WoT is proposed in a serverless NDN group chat application ChronoChat [9]. In this instance, users could cooperate to manage the chat room membership without trust anchor. Trust relationship depends on the endorsement which derived from subjective judgment of the users. This pattern of trust is not suitable for applications that with a clear definition of trust relationship, but prone to collusion attack problem.

Junjun Lou *et al.* [17] proposed a distributed NDN key management scheme based on blockchain technology. The scheme aims to publish the hash value of NDN Testbed users' public key on the blockchain. Consumers could compute the hash value of producers' public key and compare it with the hash stored in the blockchain to confirm the identity of the producer. The block height number and transaction hash value are used to locate the transactions. Iterating through every block and transaction of blockchain, we could find out the target transaction, and then the transaction needs to be parsed to obtain the data. In fact, the number of blocks and transactions is always huge, so that the iterate process could take a lot time overhead during the authentication process.

There are also some researches focus on the issues of NDN routing to improve the efficiency of content retrieval in NDN. In [29], a scheme is proposed to reduce the heavy loads on the routers which caused by huge independent Interest packets in some scenarios such as Internet of Things. Considering the mobility of the node, Zhang *et al.* [30] proposed a scheme for enhanced producer mobility support in NDN which could also reduce the network cost. Moreover, the proposed communication architecture is to support trusted communication in Flying Named Data Networking in [31].

In this section, we discussed the research of NDN content searching structures and NDN trust management schemas. We also summarized and analyzed the characteristics of each model and compare with the proposed smart contract-based content retrieval mechanism as shown in Table 3. Among them, we could see that several scenarios based on blockchain technology have additional overhead related to blockchain, such as the storage overhead and the time overhead for synchronizing the data of blockchain and traversing the blockchain transactions. The mechanism presented in this paper reduces the time overhead of traversing transactions by establishing a content name searching framework, and the storage of transactions and the cost of Gas in Ethereum will

TABLE 3. The characteristic comparison of different scenarios.

The name of the scenario	Distributed architecture	Identity certificate	Content Search	Content non-repudiation	Readable namespace	Extra overhead of verification (besides NDN)
Trust management Schema [6]		√			√	
ChronoChat [9]	√				√	
NDN key management scheme based on blockchain [17]	√	√				√
Hierarchical Name Resolution Service [8]	√	√	√			√
NDNS [7]		√	√		√	√
The Smart Contract-based content Retrieval Mechanism for NDN	√	√	√	√	√	√

be reduced by introducing the distributed storage platform in the future work.

VIII. CONCLUSION

The main work of this paper is to build a smart contract-based trusted content retrieval mechanism, which provide content consumers a trustworthy name retrieval service to find out the content information they need and avoid too many invalid interest packets occupying the limited routers cache space in NDN. The mechanism also provides trustworthy support for the authenticity and integrity of content. The formal description and the correctness analysis of the mechanism is formalized modeling by Colored Petri Nets, the cost of storage and Gas in smart contracts are evaluated by deploying the Ethereum consortium blockchain. The experiments results show that the proposed mechanism is security and practicality.

The future work is mainly about two aspects, one is to reduce the additional storage and Gas cost of the proposed mechanism because of the introduction of Ethereum smart contracts, an alternative solution is to combine the distributed storage platforms such as IPFS, which is possible to store large amounts of data from transactions in the blockchain. The other is to combine the proposed mechanism with NDN simulation environment, and considering the principle of requesting the content from closest router, we will design the routing algorithm for the name retrieval service, make a better decision to improve the content request efficiency by selecting the appropriate NDN node to request for the data packets.

The prototype of SCTCRM can be found at https://github.com/TTSone/SCTCRM.

REFERENCES

[1] L. Zhang, D. Estrin, and J. Burke, "Named data networking (NDN) project," *Relatório Técnico NDN-0001, Xerox Palo Alto Res. Center-PARC*, vol. 157, p. 158, Oct. 2010.

[2] P. Mockapetris and K. J. Dunlap, "Development of the domain name system," in *Proc. Symp. Commun. Archit. Protocols*, 1988, pp. 123–133.

[3] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, "DoS and DDoS in named data networking," in *Proc. 22nd Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2013, pp. 1–7.

[4] A. Benmoussa, A. E. K. Tahari, C. A. Kerrache, N. Lagraa, A. Lakas, R. Hussain, and F. Ahmad, "MSIDN: Mitigation of sophisticated interest flooding-based DDoS attacks in named data networking," *Future Gener. Comput. Syst.*, vol. 107, pp. 293–306, Jun. 2020, doi: 10.1016/j.future.2020.01.043.

[5] R. Tourani, S. Misra, T. Mick, and G. Panwar, "Security, privacy, and access control in information-centric networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 566–600, 1st Quart., 2018.

[6] Y. Yu, A. Afanasyev, D. Clark, K. Claffy, V. Jacobson, and L. Zhang, "Schematizing trust in named data networking," in *Proc. 2nd Int. Conf. Inf.-Centric Netw. (ICN)*, 2015, pp. 177–186.

[7] A. Afanasyev, X. Jiang, Y. Yu, J. Tan, Y. Xia, A. Mankin, and L. Zhang, "NDNS: A DNS-like name service for NDN," in *Proc. 26th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Vancouver, BC, Canada, Jul. 2017, pp. 1–9.

[8] Z. Yu, A. Dong, X. Wei, S. Guo, and Y. Yan, "Hybrid chain based hierarchical name resolution service in named data network," in *Security With Intelligent Computing and Big-data Services (SICBS)*. Berlin, Germany: Springer, 2020, pp. 629–637.

[9] Y. Yingdi and A. Alexander, "An endorsement-based key management system for decentralized NDN chat application," UCLA, Los Angeles, CA, USA, Tech. Rep. NDN-0023, 2014.

[10] *The IBM Blockchain: Blockchain Overview*. Accessed: Mar. 1, 2020. [Online]. Available: http://www.ibm.com/blockchain/what-is-blockchain.html

[11] C. Fromknecht, D. Velicanu, and S. Yakoubov, "CertCoin: A NameCoin based decentralized authentication system," Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep. 6.857 Class project, May 2014. [Online]. Available: http://courses.csail.mit.edu/6.857/2014/files/19-fromknecht-velicann-yakoubov-certcoin.pdf

[12] L. Axon. (2015). *Privacy-Awareness in Blockchain-Based PKI*. [Online]. Available: http://goo.gl/3Nv2oK

[13] S. Matsumoto and R. M. Reischuk, "IKP: Turning a PKI around with decentralized automated incentives," in *Proc. Symp. Secur. Privacy*, May 2017, pp. 410–426.

[14] L. M. Axon and M. Goldsmith, "PB-PKI: A privacy-aware blockchain-based PKI," in *Proc. 14th Int. Conf. Secur. Cryptogr.*, 2017, pp. 311–318.

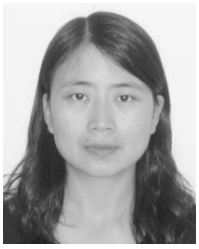
[15] M. Al-Bassam, "SCPKI: A smart contract-based PKI and identity system," in *Proc. ACM Workshop Blockchain, Cryptocurrencies Contracts (BCC)*, 2017, pp. 35–40.

[16] A. Singla and E. Bertino, "Blockchain-based PKI solutions for IoT," in *Proc. IEEE 4th Int. Conf. Collaboration Internet Comput. (CIC)*, Oct. 2018, pp. 9–15.

[17] J. Lou, Q. Zhang, Z. Qi, and K. Lei, "A blockchain-based key management scheme for named data networking," in *Proc. 1st IEEE Int. Conf. Hot Inf.-Centric Netw. (HotICN)*, Aug. 2018, pp. 141–146.
[18] Ethereum. *Blockchain App Platform*. Accessed: Nov. 28, 2017. [Online]. Available: https://ethereum.org/
[19] Solidity. Accessed: Nov. 28, 2017. [Online]. Available: https://solidity.readthedocs.io/en/develop/
[20] Z. Yan, G. Geng, H. Nakazato, Y.-J. Park, K. Nisar, and A. A. A. Ibrahim, "On-demand DTN communications in heterogeneous access networks based on NDN," in *Proc. IEEE 85th Veh. Technol. Conf. (VTC Spring)*, Sydney, NSW, Australia, Jun. 2017, pp. 1–2.
[21] A. Aboodi, T.-C. Wan1, and G.-C. Sodhy, "Survey on the incorporation of NDN/CCN in IoT," *IEEE Access*, vol. 7, pp. 71827–71858, 2019, doi: 10.1109/ACCESS.2019.2919534.
[22] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Accessed: Nov. 28, 2017. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[23] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Byzantium Version 7e819ec, Oct. 2019. [Online]. Available: https://ethereumgithub.io/yellowpaper/paper.pdf
[24] M. Baugher, B. Davie, A. Narayanan, and D. Oran, "Self-verifying names for read-only named data," in *Proc. IEEE INFOCOM Workshops*, Mar. 2012, pp. 274–279.
[25] K. Jensen and L. M. Kristensen L M, *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*. Berlin, Germany: Springer, 2009.
[26] *Truffle Suite*. Accessed: Mar. 1, 2020. [Online]. Available: https://truffleframework.com/
[27] *Go Ethereum*. Accessed: Mar. 1, 2020. [Online]. Available: https://tinyurl.com/jkw5ow9/
[28] *MetaMask*. Accessed: Mar. 1, 2020. [Online]. Available: https://metamask.io/
[29] S. Harada, Z. Yan, Y.-J. Park, K. Nisar, and A. A. A. Ibrahim, "Data aggregation in named data networking," in *Proc. TENCON-IEEE Region 10th Conf.*, Nov. 2017, p. 1.
[30] S. Zhang, Z. Yan, Y.-J. Park, H. Nakazatod, W. Kameyama, K. Nisar, and A. A. A. Ibrahim, "Efficient producer mobility support in named data networking," *IEICE Trans. Commun.*, vol. 100, no. 10, pp. 1856–1864, Oct. 2017.
[31] E. Barka, C. Kerrache, R. Hussain, N. Lagraa, A. Lakas, and S. Bouk, "A trusted lightweight communication strategy for flying named data networking," *Sensors*, vol. 18, no. 8, p. 2683, 2018.

TINGTING SONG received the bachelor's degree in software engineering from Inner Mongolia University, in 2018, where she is currently pursuing the master's degree. Her research interests include blockchain, named data networking (NDN), and privacy protecting.

BO CUI (Member, IEEE) received the M.S. degree in software engineering from Wuhan University, in 2008, and the Ph.D. degree in computer science from Inner Mongolia University, in 2017. He is currently a Lecturer with Inner Mongolia University. He has published more than ten articles in international conferences and journals. His research interests include blockchain, VANET, named data networking (NDN), and so on.



RU LI (Member, IEEE) received the M.S. degree in computer science from Peking University, in 2008, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Science, in 2005. She is currently a Professor and a Ph.D. Supervisor with Inner Mongolia University. She has published more than 50 articles in international conferences and journals. Her research interests include blockchain, wireless networks, the future Internet, and so on.

JING LIU (Member, IEEE) received the Ph.D. degree in computer architecture from the Institute of Computing Technology, Chinese Academy of Sciences, China, in 2011. He is currently a Professor of computer science and technology with Inner Mongolia University. He has published more than 20 articles in international conferences and journals. His major research interests include blockchain technology, software engineering, and formal methods.

JINSHAN SHI received the B.S. degree in computer science and technology from Inner Mongolia University, Hohhot, China, in 2009, where he is currently pursuing the Ph.D. degree. His research interests include access control issues under the Internet of Things, and using blockchain to solve problems in access control.