Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2021.013704
Article

Tech Science Press

DNS Service Model Based on Permissioned Blockchain

Yantao Shen[1,*], Yang Lu[2], Zhili Wang[1], Xin Xv[3], Feng Qi[1], Ningzhe Xing[4] and Ziyu Zhao[5]

[1]Beijing University of Posts and Telecommunications, State Key Laboratory of Networking and Switching Technology, Beijing, 100876, China
[2]Global Energy Interconnection Research Institute Co., Ltd., State Grid Office Area, Beijing, 102209, China
[3]State Grid Chongqing Electric Power Co., Electric Power Research Institute, Chongqing, 401123, China
[4]State Grid Jibei Electric Power Company, Information & Communication Company, Beijing, 100053, China
[5]The University of Western Australia, Department of Computer Science and Software Engineering, Perth, 6009, Australia
[*]Corresponding Author: Yantao Shen. Email: huakaiwuxv@163.com
Received: 13 August 2020; Accepted: 13 October 2020

Abstract: With the continuous development of the Internet, the domain name system (DNS) as infrastructure is playing an increasingly important role. However, traditional DNS architecture is centralized, and there are some security problems such as the right concentration and power abuse. This paper combines blockchain technology with DNS technology and proposes a domain name service model based on the permissioned blockchain. At first, this paper designs a top-level domain chain (TLDChain) model to conduct consensus on block transactions and achieve decentralization of domain name service. Then, this paper introduces a data model to upload data. At the same time, to improve the efficiency of the blockchain data query, this paper proposes a data warehouse to speed up the query. Also, it designs the domain name data synchronization algorithm and consistency checking algorithm to achieve synchronization and quick consistency checking of domain name data. At last, this paper introduces the domain name resolution process of this model. Experimental results show that the proposed DNS service model not only realizes the decentralization of domain name resolution but also improves query efficiency.

Keywords: Blockchain; DNS; smart contract; query optimization

1 Introduction

In the early days of Internet development, a variety of servers, routers, and switches constitute a basic network structure. And they communicate with each other through IP addresses. However, the IP address is difficult to remember. Later, DNS not only makes it convenient for people to surf the Internet but also greatly promotes the development of the Internet. The current situation is that DNS is highly centralized. There are some problems, such as right concentration, right abuse, and an unbalanced distribution of root servers and mirror servers. The Internet is positioned as an open and neutral platform. Centralized DNS management is not conducive to the development of the Internet [1,2]. Therefore, we need a scheme to solve this problem and provide a credible service for domain name resolution.

On the other hand, since the publication of Satoshi Nakamoto's white paper "Bitcoin—A Peer-to-Peer Electronic Cash System" in 2008, blockchain theory has gradually gained the attention of researchers all over the world. A new research hotspot is the use of blockchain technology in DNS service to achieve the decentralization of domain name resolution [3].

Therefore, a DNS service model based on the permissioned blockchain is proposed. The main work is to realize the decentralization of top-level domain name access by establishing a top-level domain name chain. This paper also designs the process and optimizes the domain name query. The contributions of our work are as follows:

- The DNS service model based on the permissioned blockchain is proposed, and the TLDChain model is designed for consensus of block transactions.
- The model of block data is constructed, and the uploading process of data based on the smart contract is designed.
- In the process of querying blockchain data, the synchronization algorithm and data consistency verification algorithm are designed to shorten the time of data verification.
- The domain name resolution process of this model is described, and the feasibility analysis is done through experiments.

The rest of this paper is organized as follows. Section 2 explains the existing research works related to this paper. Section 3 introduces the DNS service model based on the permissioned blockchain technology and some details in the model. Finally, the simulation analysis and the conclusion are presented in Sections 4 and 5.

2 Related Work

In the study of decentralization DNS, some scholars have already carried out some related research work.

Namecoin [4,5] is a decentralized domain name service system, which combines the public blockchain and DNS to build a system on the existing Bitcoin blockchain. The operation of Namecoin depends on lots of computing power of the Bitcoin network and manages the bit top-level domain name. However, this method will cause problems such as cybersquatting.

Blockstack [6] is divided into two parts: the underlying blockchain and the naming system. The underlying blockchain is used to record changes in name-value pairs and make consensus. The naming system performs data registration, update, and transmission. In this way, control and storage are well separated. But it is based on the Namecoin and requires lots of computing power. Therefore, it is not suitable for DNS.

BlockZone [7] proposes a distributed DNS storage and resolution scheme based on blockchain. The architecture is divided into three layers, including the user layer, storage layer, and blockchain layer. BlockZone uses the DNS server as a node in the blockchain network, allowing a node to store all the record information. However, this will consume a lot of storage.

Reference [8] classifies and describes blockchain-based DNS solutions in detail. Reference [9] proposes a new blockchain-based data block technology. It provides an advantage in data tampering. Reference [10] provides a solution to the current DNS vulnerability.

The above schemes attempt to integrate the blockchain technology into DNS, which has great reference significance for us. Considering that the public chain will consume a lot of computing power, and the addition of DNS requires a license, the permission chain is used to build the model.

3 DNS Service Model

3.1 Model Structure

The permissioned blockchain-based DNS service model (PBBDNS) is shown in Fig. 1. In this model, the top-level domain name server is responsible for the uploading of domain data and the resolution of domain name. There are three parts in this model: Data publish, TLDChain, data access.

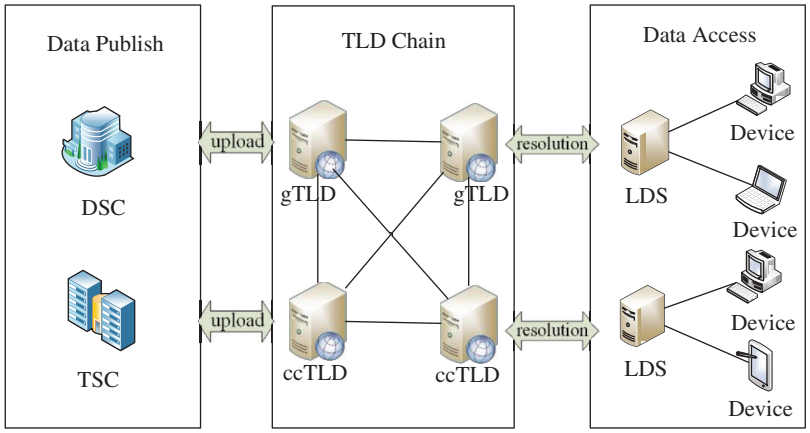

Figure 1: DNS service model based on permissioned blockchain

1) Data Publish: The uploading of domain name data

Domain Name Service Center (DSC). In the model, the DSC is responsible for the permission of the TLDChain blockchain node, providing distributed service of the top-level domain name, and preventing domain name conflict. Besides, the DSC does not participate in the consensus synchronization of domain name data.

TLD Service Centers (TSC). After the blockchain is constructed, each top-level domain name server node will register the domain name. At this time, each of the TSC needs to conduct data review permission for uploading the domain name registration information. After that, if there is a change in the information of each top-level domain name, the domain name operations, such as update and cancel, are performed according to the specific situation.

2) TLDChain: Responsible for the consensus synchronization of the domain name data

For now, the top-level domain names in DNS are divided into two categories [11], Generic Top-Level Domains (gTLDs) and Country Code Top-Level Domains (ccTLDs). This paper uses a series of gTLD and ccTLD servers in the existing domain name service system to form a permissioned blockchain, called TLDChain (Top-Level Domain Chain).

A top-level domain name server stores a table of synchronized domain names and IP addresses on the blockchain, including the data information of each top-level domain name [12].

3) Data Access: Complete resolution of domain name

Local Domain Name Server (LDS). Like the existing local domain name server, it accepts the user's domain name resolution request.

Device. Various types of users use different devices to query domain name data information that they want to access through the browser's web address input function.

3.2 Domain Name Upload

This section is divided into two parts. The first part is the block data model, which defines the data structure of the TLDChain in our model. The second part designs the process of uploading the domain name data based on smart contracts [13,14].

1) Block data model

In terms of the block data model, considering some subsequent operations, the block transaction data model is defined as shown below in Tab. 1.

Table 1: The data model of block transaction

Name	Field definition	Description
Domain Name	Top_Level_Domain	Primary key, top-level domain
Domain name status	Domain_Status	Using or cancelled
…	…	Other fields of this domain name
Hash of domain content	Domain_Hash	Current domain content hash
Global hash	Global_Hash	Global hash
History record	Historys	History of current domain name

In the block data model of the domain name, some basic information of the domain name is recorded. Besides, to trace the source of the domain name data, it is designed to add DNS history records for facilitating the quick search of the domain name history records. What is more, because the hash algorithm has a good effect on verifying consistency [15], this paper sets a hash value field in each domain name transaction record to save the hash value of this record and it uses the hash algorithm to calculate.

$$\text{domainHash} = \text{HASH}(\text{domainNameMessage}) \tag{1}$$

Then, this paper additionally sets up a global hash transaction record and places the overall hash value using the latest message hash value of each top-level domain name. It can be used for a rapid consistency checking of data.

$$\text{globalHash} = \text{HASH}(\Sigma \text{domainHash}_i)(1 \leq i \leq n) \tag{2}$$

Here n represents the number of top-level domain names.

2) Domain name data uploading

The main process of uploading domain name data to blockchain is as follows in Fig. 2.

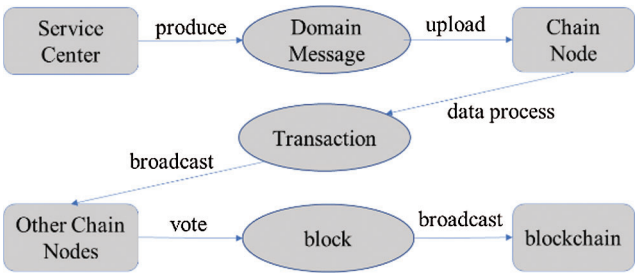

Figure 2: The process of uploading domain name to blockchain

The DSC generates the domain name message and loads it to the corresponding chain node. Before the domain name data is uploaded to the blockchain, the preprocessing of the data must be performed to make it conform to the block transaction data model designed in this paper. This paper adds a DomainStatus field for the data model to mark whether the domain name is in use or canceled.

3.3 Domain Name Query

The TLDChain stores a lot of data, and its database is based on LevelDB. LevelDB has very high random writing, and sequential reading/writing performance, but random reading performance is very

general. With the characteristics of fast writing, it has a slow query ability [16]. It is relatively time-consuming to search, which is not conducive to realtime resolution in the domain name scenario. Therefore, we design the process of domain name data query in PBBDNS as follows.

The work here is mainly divided into three parts. The first part is to design the blockchain domain name data warehouse. The second part is to synchronize data between the blockchain and blockchain domain name data warehouse and to verify data consistency using the smart contract. The last part is to realize the query of domain name data through API designed.

1) Blockchain domain name data warehouse

To improve query performance, a blockchain domain name data warehouse is set up. For the warehouse, a relational database is used. The friendly support for querying in relational databases is beneficial to the subsequent domain name query operation. In the blockchain domain name data warehouse, not lots of information about the blockchain itself is stored, but the domain name information we mainly focus on and the data of each top-level domain name is stored in a database table.

2) Domain name data synchronization and consistency check

The process of domain name data synchronization and consistency verification based on smart contract is shown in Fig. 3. In the domain name data synchronization as Algorithm 1, the smart contract is used to read updated data by the trigger conditions. Using the incremental synchronization method, data synchronization is done by block height, which does not affect the business system. Afterward, the domain name data is synchronized to the domain name data warehouse according to the different operations of the domain name.

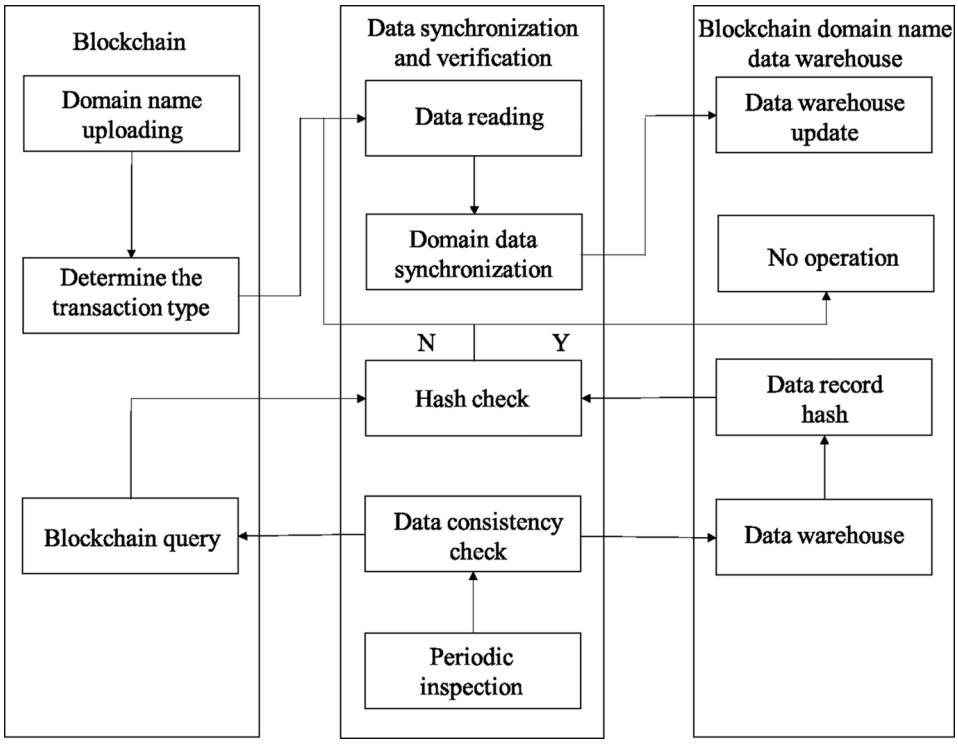

Figure 3: Domain name data synchronization and consistency check based on smart contract

Algorithm 1: Domain name data synchronization

Input: currentBlockHeight, originalBlockHeight, blockRecord
Output: result
1: **if** ∃ currentBlockHeight<originalBlockHeight **then**
2: return result ← false with BlockHeight error
3: **else if** ∃ currentBlockHeight==originalBlockHeight **then**
4: return result ← synchronized, no action required
5: **else**
6: **for** each blockheight in range(originalBlockHeight, currentBlockHeight) **do**
7: **if** blockRecord.get(option)==register **then**
8: sql.insert(blockRecord)
9: **else if** blockRecord.get(option)==update **then**
10: sql.update(blockRecord)
11: **else if** blockRecord.get(option)==cancel **then**
12: sql.delete(blockRecord)
13: **else**
14: return result ← false with error option
15: **end if**
16: **end for**
17: **end if**
18: return result ← domain name data synchronization succeeded

Blockchain domain name data warehouse has a problem of data loss due to system failure or data deletion, which needs to be checked for consistency and resynchronization. In the consistency check, the domain name data records are hashed and compared with the hash value found in the blockchain. The Algorithm 2 is a consensus algorithm based on the smart contract.

Algorithm 2: Domain name data consistency

Input: currentBlockHeight, originalBlockHeight, blockRecord
Output: result
Require: flag ← 0
1: temp ← hash(datas)
2: **if** validate(temp) is right **then**
3: continue with the datas is consistent
4: **else**

Algorithm 2 (continued).

```
flag++
for each data in datas do
    tmp ← hash(data)
    if validate(tmp) is right then
        continue with the data is consistent
    else
        data ← blockRecord
        sql.update(data)
    end if
end for
end if
return result ←flag
```

3) Domain Name Query API

The domain name data query provided externally is based on the blockchain domain name data warehouse. When designing the API interface, the data warehouse cache can be used to speed up the data query.

In the query, the corresponding domain name query data will be obtained according to one or more fields to meet the needs of the query. If the query data does not exist in the query layer cache, it will query and return results from the domain name data warehouse, and synchronize the obtained results to the query layer cache.

3.4 Domain Name Resolution

In the model of this paper, the domain name resolution process has changed due to the addition of TLDBlockchain. When the local domain name server cannot find the cache locally, it will go to the corresponding top-level domain name server in TLDChain to request information, and then return the response. It is mainly divided into six stages. Fig. 4 is the process of the domain name resolution. We will introduce the following for each stage.

(1) When a user visits a website, first he will go to the host file of his browser to find the local domain name IP cache. If IP is found, it will return a response. Otherwise, it will enter the second step.

(2) If the domain name mapping cannot be queried locally, it will request a query to LDS.

(3) The LDS will search in its DNS cache. If the domain name mapping can be found, it will return the same way. If not, it will go to the fourth step.

(4) The LDS queries the corresponding top-level domain name server. If the query is the IP of the corresponding top-level domain name, the top-level domain name server directly returns the second-level domain name server IP and enters the sixth step. Otherwise, the query obtains the top-level domain name server IP of the domain name to be queried and proceeds to the fifth step.

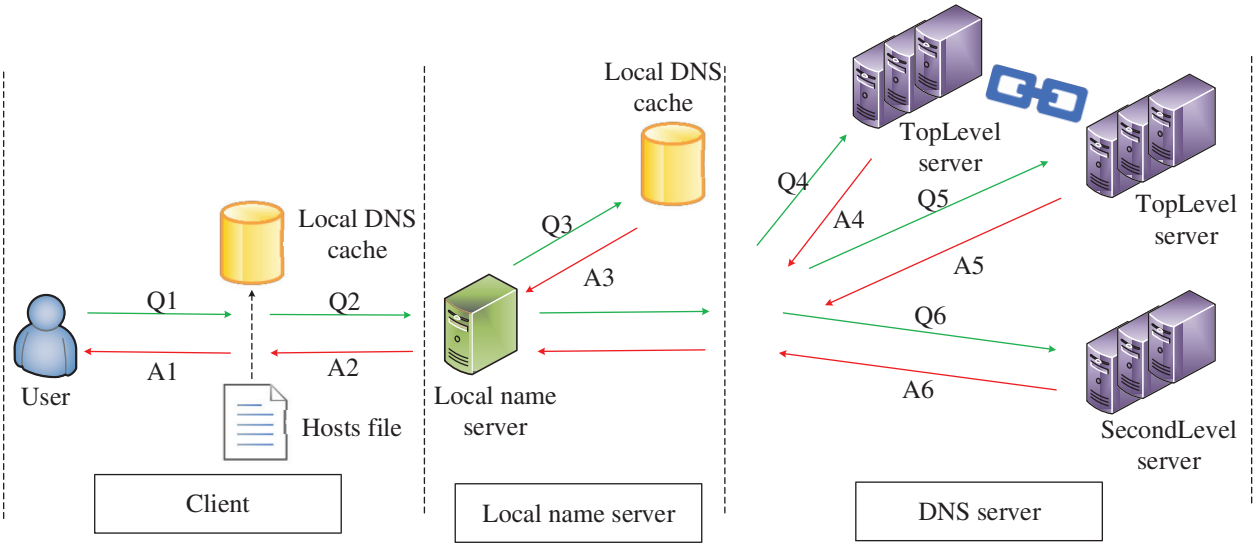

Figure 4: Domain name resolution based on permissioned blockchain

(5) The LDS sends a request to the top-level domain name server to which the query domain name belongs. And the server will return the second-level domain name server IP. Then enter the sixth step.

(6) The LDS sends a query request to the second-level domain name server to which the query domain name belongs and then performs an iterative query operation.

4 Experimental Simulation

This paper defines the experimental environment as follows. The operating system is Ubuntu Server 18.04 LTS 64-bit. The computer hardware parameters are one core, 2G memory, and 50G hard disk. Hyperledger Fabric v1.4 version is used as the simulation blockchain. Go v1.10 version is used to write the chain code and system implementation, and MySQL v5.7 version is used as the database.

In terms of consistency check, the consistency check is performed when there are 100 to 1000 domain names in the blockchain in Fig. 5.

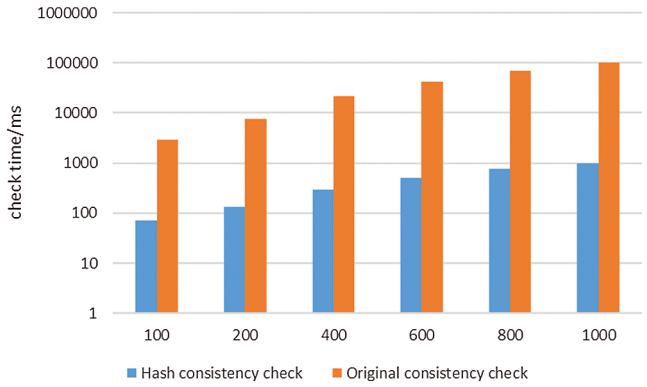

Figure 5: Time consumption comparison graph of data consistency check

It can be seen that with the increase of the domain names in the blockchain, the time required for consistency check is on the increase. But at the same time, we can also see that the newly designed hash consistency check scheme greatly reduces the consistency check time.

The direct query in the blockchain and the query optimization was performed in Fig. 6. It can be seen that the designed process greatly improves the query speed.

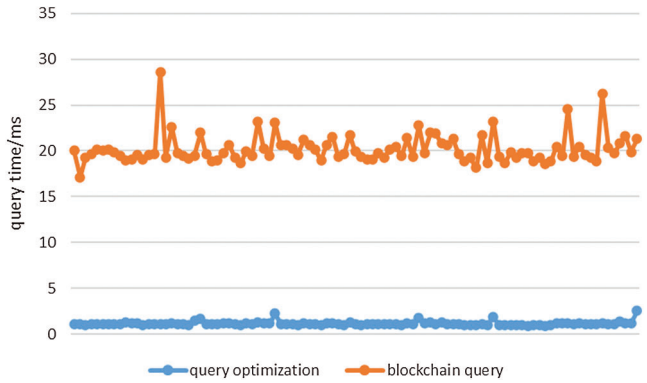

Figure 6: Comparison chart of domain name data query

The domain name resolution experiment mainly carried out the resolution test for a second-level domain name. In the experiment, a second-level domain name resolution operation is performed. The time-consuming comparison is in Fig. 7.

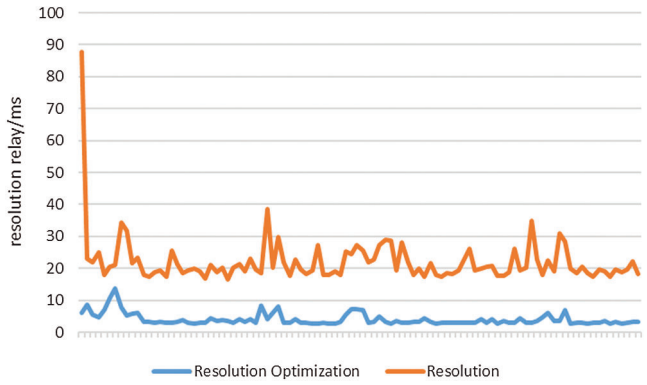

Figure 7: Comparison graph of second-level domain name resolution delay

Compared to directly querying data in the blockchain, the optimization process proposed in this paper greatly improves the speed of domain name resolution.

5 Conclusion

The traditional domain name service system has a centralized status, which is not conducive to the openness and equality of the Internet. In this paper, a DNS service model is proposed, and the TLDChain is designed for consensus of block transactions. The paper realizes domain name uploading, query optimization, and resolution based on the smart contract. Experimental data shows that the DNS service

model proposed in this paper not only realizes the decentralization of domain name resolution but also solves the problem of query efficiency caused by the blockchain.

In future work, we will devote ourselves to the design of consensus algorithms and the improvement of models.

Funding Statement: This work has been supported by State Grid Corporation of China science and technology project "Research on Reliable Transmission Technology in Electric Internet of Things Based on IPv6" (5700-202058178A-0-0-00).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. Zhang, C. D. Xia, B. X. Fang and H. L. Zhang, "An autonomous open root resolution architecture for domain name system in the internet," *Journal of Cyber Security*, vol. 2, no. 4, pp. 57–69, 2017.

[2] P. P. Yuan and C. Q. Wang, "Research on DNS security threats and countermeasures," *Cyberspace Security*, vol. 9, no. 5, pp. 50–54, 2018.

[3] T. S. Zhuang, W. F. Liu and D. Li, "DNS root domain name analysis system based on block chain," *Telecommunications Science*, vol. 34, no. 53, pp. 17–22, 2018.

[4] Martin Haferkorn and J. M. Q. Diaz, "Seasonality and interconnectivity within cryptocurrencies-an analysis on the basis of bitcoin, litecoin and namecoin," in *Proc. FinanceCom 2014*, Sydney, SYD, Australia, pp. 621–632, 2015.

[5] W. H. Hu, M. Ao, L. Shi, J. G. Xie and Y. Liu, "Review of blockchain-based DNS alternatives," *Chinese Journal of Network and Information Security*, vol. 3, no. 3, pp. 71–77, 2017.

[6] M. Ali, J. Nelson, R. Shea and M. J. Freedman, "Blockstack: A global naming and storage system secured by blockchains," in *Proc. USENIX ATC 16*, Denver, CO, USA, pp. 181–194, 2016.

[7] W. T. Wang, N. Hu and X. Liu, "Blockzone: A blockchain-based DNS storage and retrieval scheme," in *Proc. ICAIS 2019*, New York, NY, USA, pp. 155–166, 2019.

[8] K. Enis and E. Adiguzel, "Blockchain based DNS and PKI solutions," *IEEE Communications Standards Magazine*, vol. 2, no. 3, pp. 52–57, 2018.

[9] W. Yoon, I. Choi and D. Kim, "BlockONS: Blockchain based object name service," in *Proc. ICBC*, Seoul, SEL, Korea (South), pp. 219–226, 2019.

[10] B. Benshoof, A. Rosen, A. G. Bourgeois and R. W. Harrison, "Distributed decentralized domain name service," in *Proc. IPDPSW*, Chicago, CHI, USA, pp. 1279–1287, 2016.

[11] D. X. Nan, W. Wang, Z. W. Yan and G. G. Geng, "A domain name management architecture model based on blockchain," *e-Science Technology & Application*, vol. 10, no. 4, pp. 19–29, 2019.

[12] J. Q. Liu, B. Li, L. Z. Chen and M. Hou, "A data storage method based on blockchain for decentralization DNS," in *Proc. DSC*, Guangzhou, GZ, China, pp. 189–196, 2018.

[13] C. G. Ma, J. An, W. Bi and Q. Yuan, "Smart contract in blockchain," *Netinfo Security*, vol. 18, no. 11, pp. 8–17, 2018.

[14] L. W. Ouyang, S. Wang, Y. Yuan, X. C. Ni and F. Y. Wang, "Smart contracts: Architecture and research progresses," *Acta Automatica Sinica*, vol. 45, no. 3, pp. 445–457, 2019.

[15] C. Su, "Block chain optimization model based on consistent hash algorithm," *Computer Knowledge and Technology*, vol. 15, no. 14, pp. 163–165, 2019.

[16] H. J. Wang, B. R. Dai, C. Li and S. H. Zhang, "Query optimization model for blockchain applications," *Computer Engineering and Applications*, vol. 55, no. 22, pp. 34–39, 2019.