Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2021.016839

ARTICLE

IDV: Internet Domain Name Verification Based on Blockchain

Ning Hu[1], Yu Teng[2], Yan Zhao[1], Shi Yin[1] and Yue Zhao[3,*]

[1]Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, 510006, China

[2]Zhong Zi Hua Ke Traffic Construction Technology Co., Ltd., Beijing, 100195, China

[3]Science and Technology on Communication Security Laboratory, Chengdu, 610041, China

[*]Corresponding Author: Yue Zhao. Email: yuezhao@foxmail.com

Received: 11 March 2021 Accepted: 24 June 2021

ABSTRACT

The rapid development of blockchain technology has provided new ideas for network security research. Blockchain-based network security enhancement solutions are attracting widespread attention. This paper proposes an Internet domain name verification method based on blockchain. The authenticity of DNS (Domain Name System) resolution results is crucial for ensuring the accessibility of Internet services. Due to the lack of adequate security mechanisms, it has always been a challenge to verify the authenticity of Internet domain name resolution results. Although the solution represented by DNSSEC (Domain Name System Security Extensions) can theoretically solve the domain name verification problem, it has not been widely deployed on a global scale due to political, economic, and technical constraints. We argue that the root cause of this problem lies in the significant centralization of the DNS system. This centralized feature not only reduces the efficiency of domain name verification but also has the hidden risks of single point of failure and unilateral control. Internet users may disappear from the Internet due to the results of fake, subverted, or misconfigured domain name resolution. This paper presents a decentralized DNS cache verification method, which uses the consortium blockchain to replace the root domain name server to verify the authenticity of the domain name. Compared with DNSSEC's domain name verification process, the verification efficiency of this method has increased by 30%, and there is no single point of failure or unilateral control risk. In addition, this solution is incrementally deployable, and even if it is deployed on a small number of content delivery network servers, satisfactory results can be obtained.

KEYWORDS

Blockchain-based network security; DNS security; DNS decentralization; CDN

1 Introduction

The rapid development of blockchain technology has provided new ideas for network security research. Due to the characteristics of decentralization, anti-tampering of data, and traceability of transactions, blockchain technology is widely used in Internet infrastructure [1–3], information sharing [4], privacy protection and other aspects. In fact, blockchain-based network security reinforcement solutions have become a hot topic of widespread concern in the academic community [5–7].

The DNS (Domain Name System) is a crucial infrastructure of the Internet, providing key support services for the normal operation of various naming service-based web applications, emails, and distributed systems. The security of the DNS is of vital importance to the stable operation of the Internet [8]. Although DNS has been improving and perfecting since it was proposed, frequent security incidents in recent years have shown that there are still many challenges to DNS security that need to be resolved [9,10]. We argue that the management model of a DNS has a typical centralization feature, which weakens the working efficiency and security protection capabilities of the DNS to some extent [11]. The research results in recent years show that the decentralization of DNS will help improve the robustness and security of the DNS [12,13]. The emergence of blockchain technology has proposed a new direction for the research of this problem.

Since Kaminsky proposed a DNS cache poisoning attack in 2008 [14], ensuring the authenticity of domain name resolution results has always been the key to DNS security. To effectively prevent DNS buffer poisoning attacks, DNSSEC (Domain Name System Security Extensions) and NSEC3 (Next Secure version 3) are introduced into the DNS [15]. DNSEC is based on public key cryptography. DNS resource records calculate digital signatures and verify the authenticity of the resolution results. Although DNSSEC can eliminate the problem of fake domain names well in theory, its verification process relies on the root of trust, which has typical centralized characteristics. Since all domain name verification must go through the root of trust, there is a risk of unilateral control. As an alternative to DNS, DNSSEC has no gradual deployment capability. For the above reasons, DNSSEC is currently only deployed at the top-level domain name level and cannot provide authenticity guarantees for second-level domain names. Source port randomization is another solution to prevent terminals from DNS cache poisoning attacks [16]. By dynamically changing the source port of the DNS resolution request, it makes it difficult for attackers to forge false response information. Unfortunately, such methods cannot completely prevent the occurrence of DNS cache poisoning attacks. For example, Alharbi et al. [17] proposed a DNS cache poisoning method, and it was successfully tested on Windows, Mac OS, and Ubuntu Linux, which further showed that the cache poisoning attack could not be prevented by port hopping.

Inspired by the principle of CDN (Content Delivery Network) acceleration, we propose a CDN-based authenticity verification method for domain names [18]. As we know, CDN refers to a geographically distributed group of servers that work together to provide fast delivery of Internet content. To improve the surfing experience of users, ISPs will deploy many CDN servers at the edge of the network, and these servers provide Internet users with website content caching services. Because these cache servers are close to user terminals, they can provide a more lightweight domain name verification service than DNSSEC. Considering the great difficulty in widely deploying new protocols such as DNSSEC, we designed IDV (Internet Domain-name Verification) as a separate protocol that does not change the resolution procedure and protocol of traditional DNS. The core ideas of this method are as follows. First, we deployed a domain name verification daemon on the DNS client. The daemon can actively detect changes in the local DNS cache. When the new domain name resolution result appears in the DNS cache, the daemon will send a domain name verification request to the CDN server. Second, the CDN server forms a consortium blockchain to jointly ensure the authenticity of the domain name. When the CDN receives a domain name verification request, it verifies the authenticity of the domain name through the smart contract deployed on the blockchain and returns the verification result. Since CDN servers are distributed all over the world and are independent of each other, this greatly

increases the difficulty of DNS cache poisoning attacks and reduces the possibility of passive fraudulent actions on CDN servers. In addition, with the help of smart contract technology, the credibility of the domain name verification code execution process can be guaranteed. Finally, the CDN server not only provides domain name authenticity verification but also attracts more users to use its Internet access acceleration service. Therefore, this method has good incentives.

The main contributions of this paper are described as follows:

1) We present a decentralized DNS cache verification method, which uses the consortium blockchain to replace the DNS root server to verify the authenticity of the domain name. Compared with DNSSEC's domain name verification process, our verification efficiency has increased by 30%, and there is no single point of failure or unilateral control risk. In addition, our solution is incrementally deployable, and even if deployed on a small number of CDN servers, satisfactory results can be achieved. Moreover, such deployment does not interfere with the operation of non-participants.
2) We propose a lightweight domain name authenticity verification algorithm, which can be used by multiple CDN servers to verify the authenticity of domain names. Our algorithm has a faster consensus speed and lower network traffic, which can ensure that the domain name verification process is completed in a short time.
3) We designed and proposed a prototype that can be deployed gradually in the Internet environment. We also conducted experiments to verify the effectiveness of the method. The experimental results show that the domain name verification process can usually be completed within 1 s. Therefore, the method proposed in this article will not cause too much trouble to the user experience.

The remainder of this paper is organized as follows: Section 2 discusses related studies and analyses their limitations. Section 3 describes the architecture and algorithms of our solution in detail. Section 4 presents the results of experiments and evaluations. In Section 5, IDV and DNSSEC are comparatively analysed in terms of effectiveness, credibility and incentives. Section 6 presents the conclusions.

2 Related Work

Although the Internet DNS is a physically distributed system, it has significant centralization features in many aspects, such as domain name space organization, domain name resolution, and domain name verification. At present, there are 13 root zone servers in the world, mainly distributed in 4 countries, including the United States and Japan. This kind of overconcentrated architecture also has system availability problems and serious unilateral control risks [19]. Simply improving the DNS protocol or implementation technology cannot solve these problems well. For this reason, DNS security solutions based on decentralization have recently become a hot topic.

2.1 Blockchain-Based Security Enhancement Solution

To change the centralized management model of DNS, many decentralized DNS implementation schemes have been proposed. In the decentralized DNS, the management of domain name resources no longer relies on the centralized organization and root servers, so the domain name resolution and verification process are more efficient, and there is no unilateral control risk.

Namecoin [20] is a new type of Bitcoin-based DNS system. In Namecoin the transaction information stored in the blockchain is replaced with name-value mapping data. Therefore, Namecoin and Bitcoin [21] have most of the functions and mechanisms in common, but Namecoin

is a more general name-value pair resolution system, not a replacement for the current DNS. Namecoin uses different prefixes to match other types of name-value pairs. The 'id /' prefix is used for registered identities. Namecoin preserved the name '.bit' as the top-level domain name, and this domain name has not been officially registered in the current DNS. Because Namecion and DNS are two completely independent systems, although NameCoin provides a mechanism to quickly resolve and verify domain names through blockchain technology, it does not help traditional DNS.

Blockstack [22] was proposed to increase the scalability of Namecoin. Due to the limited data storage capacity of the blockchain, Blockstack stores the resource records of DNS in an external database and only stores the metadata of domain names in the blockchain ledger. Blockstack divides the DNS into a control plane and a data plane. The control plane provides domain name resolution and verification functions, and the data plane provides data storage and retrieval functions. Blockstack's data plane allows multiple service providers to provide data storage services. In addition, there are still a variety of Bitcoin derivative systems based on blockchain technology, and these systems also provide name resolution services, such as ENS [23], Peername [24] and EMCDNS [25].

There are also some results that try to build a legacy DNS on the blockchain platform. These research works generally have the following commonality. First, domain name release management is realized based on on-chain transactions, which ensures that domain name information is globally visible and overcomes the domain name blind zone problem of existing DNS. Second, the organizational structure of domain name storage is flattened through the P2P network, and any node can reserve a complete domain name space. Finally, domain name resolution does not need to be completed by multiple authoritative servers. Any host can obtain domain name resolution results by querying the blockchain ledger. For example, He et al. [26], proposed TD-Root which builds the root domain name system of DNS on the blockchain platform, which not only improves the security of the root domain name system but also eliminates the threat of unilateral control. In TD-Root, complete domain name publishing, updating, and query operations are provided so that users can perform domain name resolution similar to accessing a traditional DNS. Hu et al. proposed a decentralized and trustworthy data plane for DNS, which is named BlockZone [11]. In this work, the authors implement the data plane of DNS on blockchain and increase the efficiency of domain name resolution and verification. Since the DNS client resolves domain names through a trustworthy distributed ledger, it can effectively resist the DNS buffer poisoning attack. Li et al. proposed a DNS based on blockchain, called B-DNS. B-DNS provides a traditional DNS query interface to the terminal and uses the blockchain to store the resource records of the domain name at the bottom layer. To improve performance, B-DNS replaces the PoW consensus protocol with a PoS consensus protocol with higher processing performance. The experimental results of the paper show that the blockchain-based DNS is more secure than the traditional DNS.

The decentralization of the DNS based on blockchain technology can make the DNS domain name organization structure flatter, improve the efficiency of domain name resolution and verification, and reduce the risk of unilateral control of the root server. However, most of the existing decentralized DNSs have the following limitations. First, all these solutions are incompatible with the traditional DNS, so it is difficult to deploy a blockchain-based domain name system on a large scale. Second, the underlying implementations of Namecoin and Blockstack are both Bitcoin. Because Bitcoin uses the "one-CPU-one-vote" mechanism, if an organization controls 51% of the entire system's computing power, that is, a 51% attack, it will cause serious security concerns for the system and hidden danger. Although 51% attacks exist in theory, if they have 25% of the computing power, they can threaten the security of the system [27]. Last, because

the blockchain stores all historical information, the entire system will become increasingly larger, and it is difficult for mobile devices or personal computers to have enough hard disk space to store all the recorded information. Although some scholars have proposed the SNV (Simple Name Verification) protocol [28], it is necessary to set up a server that provides all records, and the communication security between the server and the client is another problem that needs to be solved.

2.2 Monitoring-Based Security Enhancement Solution

Since the traditional DNS cannot be replaced by a new system in a short time, many DNS security enhancement solutions based on monitoring have been proposed. These solutions do not need to change the original implementation of the DNS but determine potential abnormal behaviours by monitoring the protocol traffic of the DNS or the resolving results.

Madariaga et al. proposed a DNS anomaly detection method called AD-BoP [29]. This method first uses a large amount of historical data to train machine learning algorithms and then uses the trained algorithm to detect anomalies in the traffic of top-level domain (TLD) name servers. In the traditional DNS domain name resolution process, once the TLD domain name is attacked, many domain names will be inaccessible. Therefore, monitoring the traffic of TLD registry operators can effectively improve the reliability of the DNS. Lyu et al. propose a distributed DNS monitoring method that tracks DNS traffic at the three levels of the host, sub-net and autonomous system and comprehensively analyses the monitoring results of the three levels [30]. This method can help the victim isolate the source of the attack.

Since the main target of the DNS buffer poisoning attack is the client that sends the domain name resolution request, improving the security protection capability of the client can also help solve the security problem of DNS. Maksutov et al. [31] propose a collaborative verification idea for the pharming attack. When the client needs to resolve the domain name, it sends requests to multiple servers at the same time. When the client receives the response returned by the server, it compares these responses to determine the authenticity of the returned result. This method has certain limitations. First, the impact of CDN on domain name resolution is not considered. To speed up the user's access to the website, some large CDN networks will deploy many servers in the network. These servers will cache many domain name resolution results and return the nearest server address based on the user's location. Therefore, the domain name resolution results from different servers may indeed be different. Therefore, the false positive rate of this method is relatively high.

3 IDV Architecture

3.1 Basic Idea

Despite the emergence of many research results on the reconstruction of the DNS based on blockchain technology, in the actual network environment, it is rare to see large-scale commercial deployments. Therefore, this type of work is still in its infancy. In fact, because the Internet is a complex network composed of operators from all over the world, it is difficult for operators of various countries to redeploy a new DNS in a short period of time. For example, although DNSSEC has been proposed for many years, only the root domain name is protected by DNSSEC thus far. To this end, we hope to find a DNS security solution that can be independently deployed by operators and can be incrementally improved.

Inspired by the principle of CDN acceleration, we found that the use of a CDN server can provide a local DNS domain name verification service. To provide a better surfing experience

and attract more users, CDN vendors will intercept users' DNS resolution requests and redirect user traffic to a closer buffer server based on the user's domain name resolution results, thereby reducing network access latency. Today, CDN services have been widely deployed on a large scale globally. Tab. 1 lists the current global server deployment scales of some CDN providers.

Table 1: CDN node statistics

CDN vendor	CDN nodes in worldwide
Aliyun [32]	500+
Tencent [33]	1000+
Akamai [34]	4100+
Cloudflare [35]	500+
Amazon cloudFront [36]	225
Google [37]	100+
Microsoft azure [38]	500+

We found that CDN services have good progressive deployment capabilities. Only a small number of CDN cache servers need to be deployed on the edge network to obtain a good network access acceleration effect, and as the number of servers increases, the acceleration effect will continue to grow. By using the CDN server, we propose a method of DNS domain name verification, called IDV. IDV consists of a set of verification services deployed on the CDN cache server and domain name verification clients deployed on user terminals. The verification client runs as a background process on the DNS client host and is responsible for monitoring the local DNS cache. Once the verification client finds that a new domain name resolution result appears in the local DNS cache, it immediately sends a domain name verification request to the verification server. When the CDN cache server receives the domain name verification request, it completes the authenticity verification of the domain name locally and returns the verification result to the client. Generally, when a terminal accesses a domain name for the first time, it needs to perform domain name resolution through DNS. According to the principle of CDN acceleration, once the terminal completes domain name resolution, the content corresponding to the domain name will be cached in the CDN cache server. Therefore, the CDN server can identify the authenticity of the domain name resolution result. In addition, to prevent a single CDN server from failing to buffer hits, IDV uses blockchain technology to organize multiple CDN cache servers into a consortium blockchain. Data synchronization between different CDN cache servers is achieved through the P2P network. By using blockchain technology, the CDN server completes the verification of the domain name through the smart contract. Since the smart contract ensures the credibility of the code execution process, it can prevent the malicious CDN server from providing false verification results.

The system architecture of IDV is shown in Fig. 1. There are four processes in Fig. 1. The first process is 'name resolving', which is a standard DNS domain name resolution process. When the user terminal needs to access a certain domain name, it first checks the local DNS cache, and if it does not hit, it sends a domain name resolution request to the nearest DNS domain name resolution server. The second process is 'DNS monitoring'. The CDN cache server monitors the DNS resolution request sent by the user terminal in real time. Once a new domain name resolution result is found, it will cache the result locally and then start the third process. The third

process is 'Result validating'. In this process, the authenticity of the domain name resolution result is determined, and if the determination is successful, the result is written into the blockchain. In the above three processes, Process 1 and Process 2 and Process 3 are carried out at the same time. Process 4 is 'Name validating'. This is the process by which the IDV client verifies the domain name resolution result. When the IDV client detects a new domain name resolution result in the local cache, it starts the process and requests the CDN server to verify the authenticity of the result.

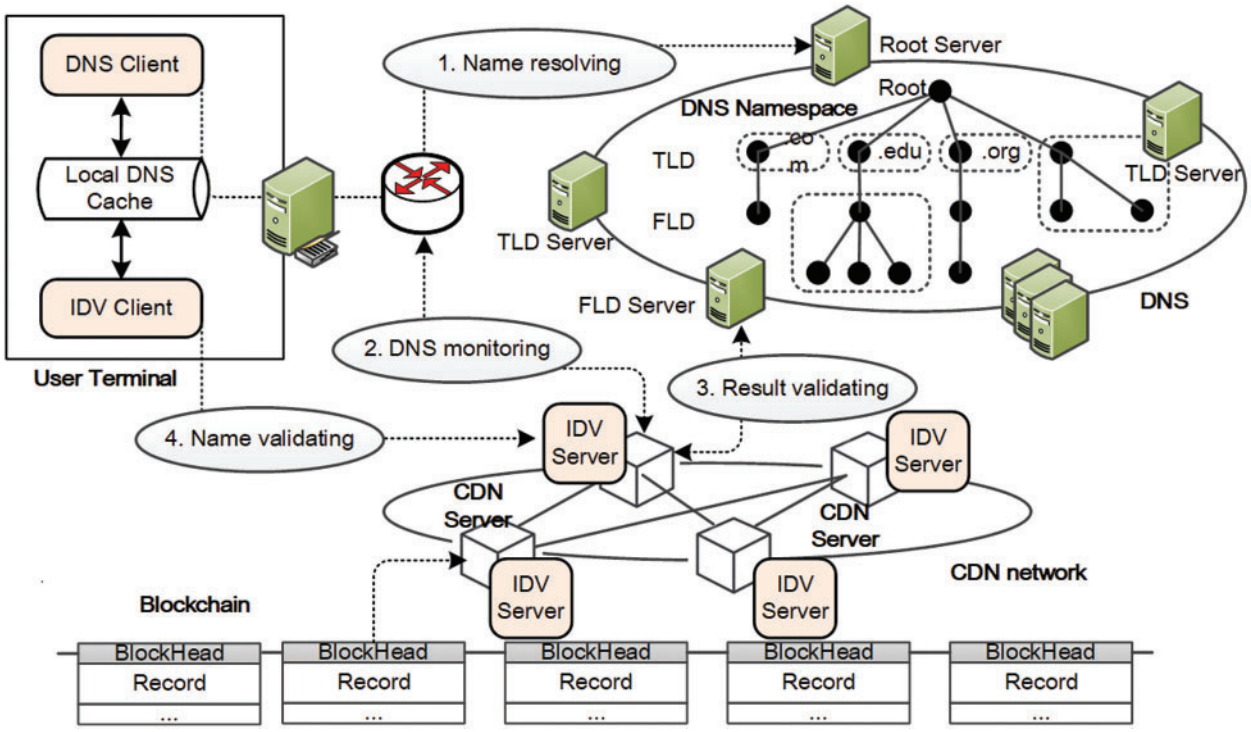


Figure 1: Architecture of IDV

3.2 Result Validating

Since 'name resolving' and 'DNS monitoring' are relatively simple and are the normal functions of the DNS and CDN service, respectively, we do not discuss them in this paper. In this section, we mainly introduce the process of result verification. In the process of design result verification, the following issues need to be considered:

To implement the process of result verification, the following issues must be considered:

(1) When the CDN cache server observes a new domain name resolution result, how should it initiate the verification process?
(2) How to save the result of domain name verification?
(3) How to ensure that the CDN cache server verifies the domain name resolution results in accordance with the specified procedures?

IDV uses the consortium blockchain to solve the above problems. In IDV, all the CDN servers are formed into a consortium chain.

First, when any CDN cache server observes a new domain name resolution result, it first checks the local ledger. If the verification result of the domain name is already recorded in the ledger, there is no need to verify it again. If the domain name verification result is not found in the ledger, the domain name resolution process is started, and the domain name resolution result is verified through the DNS authoritative server. This process can be either a recursive process of DNS or an iterative process. After the result is confirmed, an accounting transaction request is initiated, and different CDN cache servers determine the accounting order and accounting nodes through a consensus protocol. For the second issue, IDV directly writes the verification results of the domain name and verification nodes as transaction records into the blockchain ledger and uses the P2P network of the blockchain to achieve data synchronization across the entire network. In this way, other CDN nodes can query the ledger and quickly confirm the authenticity of the domain name. For the third issue, IDV uses smart contracts to implement the domain name verification process. Because the smart contract code cannot be easily changed once it is written into the blockchain, it can ensure that the domain name verification process will not be tampered with. Finally, IDV adopts a consortium chain method. Therefore, only CDN cache nodes have the authority to keep accounts. Therefore, it can prevent malicious terminals from imitating CDN cache server nodes to keep malicious accounts. According to the above description, the algorithm of the 'Result validating' process is described as follows. The parameter and function descriptions of Algorithm 1 are listed in Tab. 2.

Algorithm 1: 'Result validating' process

Input: *TargetName*, *Resolving_Result*
Output: **Valid** or **Invalid**

```
1.  MatchSet ⇐ NULL;
2.  MatchSet ⇐ searchLocalCache(TargetName);
3.  If MatchSet == NULL then {
4.      MatchSet ⇐ searchLedger (TargetName);
5.   If MatchSet == NULL then {
6.      R_Result ⇐ InvokeNameResolving(TargetName);
7.      If Resolving_Result == R_Result then {
8.         InvokeTransaction(TargetName, Resolving_Result, Valid);
9.         UpdateLocalCache(TargetName, Resolving_Result, Valid);
10.        return Valid;
11.    }
12.    else {
13.        InvokeTransaction(TargetName, Resolving_Result, InValid);
14.        UpdateLocalCache(TargetName, Resolving_Result, InValid);
15.        return InValid;
16.    }
17. else {
18.    for (each item in MatchSet) {
19.       if item. Resolving_Result == Resolving_Result then
20.          return Valid;
21.    }
```

(Continued)

Algorithm 1: (Continued)

```
22.   InvokeTransaction(TargetName, Resolving_Result, InValid);
23.   UpdateLocalCache(TargetName, Resolving_Result, InValid);
24.   return InValid;
25. }
```

Table 2: Parameter and function descriptions of Algorithm 1

Symbol	Description
TargetName	The domain name needs to be validated
Resolving_Result	Resource records of target domain name {'A', 'NS','CNAME','SOA', ... }
MatchSet	Used to store matching result.
searchLocalCache( )	To improve efficiency, each CDN cache server maintains a domain name verification list in memory and uses the least recently used elimination (LRU) algorithm.
searchLedger ( )	Searches the blockchain ledger.
InvokeTransaction( )	Constructs a transaction request and submits a transaction request to the endorsing node.
UpdateLocalCache( )	Writes the latest verification result to the local Cache

3.3 Name Validating

To accelerate access to network resources, most operating systems and browsers will cache the DNS resolving result and share it with other applications. Unfortunately, neither the operating system nor the browser itself can verify the authenticity of the domain name resolution results. Therefore, when the terminal suffers a DNS cache poisoning attack or the terminal obtains the domain name resolving result from a server that has suffered a DNS cache poisoning attack, the applications on this terminal will use the fake IP address. Research results show that any level of DNS cache is subject to a DNS cache poisoning attack [17,39].

The IDV client is installed as a system service to run on the user terminal. To protect the OS-wide DNS cache, the IDV client periodically scans the local cache. Once a new domain name resolving result is found, the domain name verification process will be triggered. When a fake domain name is found, the IDV client generates a system alarm and cleans the OS-wide DNS cache.

The monitoring algorithm of the DNS local cache is shown in Algorithm 2, and the descriptions of some parameters and functions are listed in Tab. 3.

3.4 Implementation and Deployment

When IDV is implemented, the following issues need to be considered:

(1) Trustworthy. The verification result of the domain name needs to be used multiple times, so it is necessary to ensure the credibility of the verification result, and the result cannot be easily tampered with during the storage process.

(2) Sequentially. The correspondence between domain names and IP addresses may change dynamically. Therefore, the domain name verification results generated by different IDVs need to maintain a uniform time sequence in the global scope.
(3) Consistency. When different IDV servers verify the authenticity of the domain name resolution results, they may use different authoritative servers. In some cases, there may be multiple IDV servers that publish the domain name verification results at the same time. Therefore, it is necessary to maintain global consistency.

In addition to the above problems, IDV also needs to implement the following mechanisms.

Table 3: Parameter and function descriptions of Algorithm 2

Symbol	Description
TargetName	The domain name needs to be validated
sysCachePath	The accessing path of OS cache file
browserCachePathList	The accessing path of different browser cache files
IDVServerIP	IP address of IDV Server
scanOSCache ( )	Scans files specified by *sysCachePath* and *browserCachePathList* in turn
SearchValidationLog()	Searches the log file for the domain name verification record corresponding to the Item
cleanCache()	Removes the fake DNS resolving results from the cache
askForValidation()	Sends validation request to IDV server

Algorithm 2: 'Name validating' process

```
Input: sysCachePath, browserCachePathList, IDVServerIP, scanInterval
Output: LogItem;
1. while true do {
2.    isChanged ⇐scanOSCache(sysCachePath, browserCachePathList) ;
3.    if ( isChanged == True ) then {
4.       validationList ⇐ NULL;
5.       newItemList ⇐buildList(sysCachePath, browserCacheList) ;
6.       for (each item in newItemList) {
7.          oldItem ⇐ SearchValidationLog(item);
8.          if ( oldItem == item ) then {
9.             if isValid( item ) == FALSE then {
10.               cleanCache(item, sysCachePath, browserCacheList);
11.               Alarm();
12.            }
13.         }
14.         else
15.            append(validationList, item);
16.      } //end of for
17.      if (validationList is not NULL) then {
18.         ack ⇐askForValidation(IDVServerIP, validationList);
```

(Continued)

Algorithm 2: (Continued)

```
19.             for (each item in validationList) {
20.                 update(item, valiationLog);
21.                 if isValid(item) == FLASE then {
22.                     cleanCache(item, sysCachePath, browserCacheList);
23.                     Alarm();
24.                 }
25.             }
26.         }
27.     }
28.     sleep(scanInterval);
29. }
```

First, because IDV is deployed on the CDN network, to avoid affecting the performance of the CDN network, IDV does not require all CDN cache servers to deploy IDV. To prevent malicious verification, only certified and authorized CDN cache servers can be allowed to join the system and run the IDV server. Second, different CDN vendors may have conflicts of interest, so we cannot assume that these investors are willing to establish collaborative relationships. Therefore, the servers of different CDN investors may be divided into multiple groups and run IDV independently. Finally, to protect customer privacy, IDV servers of different groups may use different channels for domain name verification.

Blockchain is a new type of distributed system. The distributed ledger can not only keep the data persistent and ensure global consistency but also provide a smart contract mechanism to ensure the credibility of the code execution process. Therefore, the blockchain can provide a good foundation for IDV. The current blockchain technologies mainly include public chains, private chains and consortium chains. The public chain is a completely decentralized blockchain technology. Its characteristic is that anyone can join freely, and all nodes have a peer-to-peer relationship. The public chain can guarantee fairness, but it takes longer to verify the completion of the transaction. The private chain is the opposite of the public chain. Only organizations can write to the blockchain, and its scalability is weak. The consortium chain allows multiple organizations to participate, and each organization can control some nodes for transactions. According to the above analysis of IDV, the alliance chain is the most suitable choice.

We have implemented a prototype system of IDV based on Hyperledger Fabric [40]. The system composition is shown in Fig. 2.

In Fig. 2, the IDV client is deployed on the user terminal and is responsible for monitoring the DNS cache. When a new domain name resolution result is observed, Algorithm 2 is used to initiate a domain name verification request. The IDV server is deployed on the CDN cache server, and when it receives a verification request for the domain name resolution result, it uses Algorithm 1 for verification. The consensus verification part in Algorithm 1 is stored on the blockchain as the Fabric chaincode. According to the working principle of Hyperledger Fabric, all servers deployed with the IDV server are peer nodes. Through configuration, the respective working roles can be further distinguished, including endorser nodes, committer nodes and leader nodes. In addition, a separate CA centre needs to be deployed for node registration and identity management. When the IDV server needs to write a new verification record to the ledger, the processing process is shown in Fig. 3.

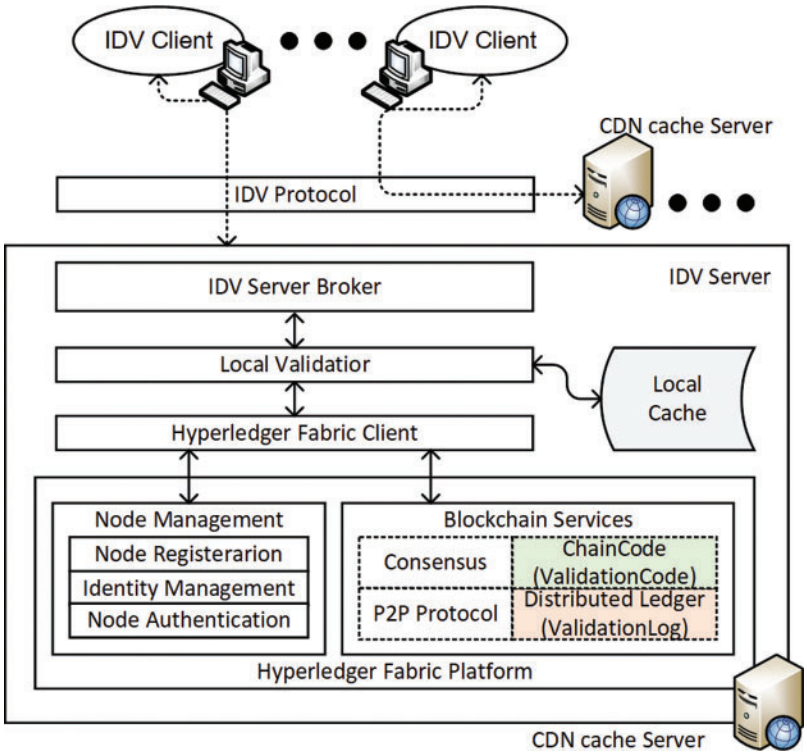


Figure 2: Composition of IDV

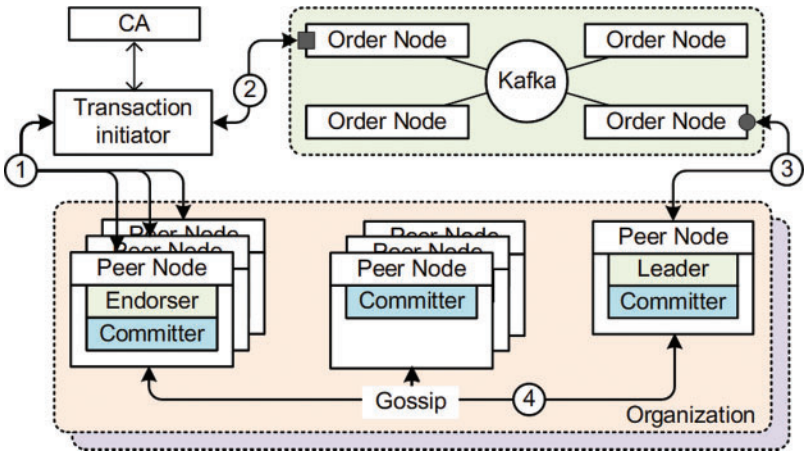


Figure 3: The transaction processing flow in Hyperledger Fabric

First, the transaction initiator constructs a message of the transaction proposal and sends the message to the endorsing node. After the endorser simulates the execution of the transaction proposal, it returns a confirmation message to the transaction initiator. When the transaction initiator collects enough confirmation from different endorsers, it submits a transaction request to the order node. After the order node completes the ordering of the transaction request, a block is generated. This process is usually done through a consensus protocol, such as Kafka or Raft.

The leader obtains the block through the order node and writes it into the ledger, and then the leader node completes the ledger synchronization with other committer nodes through the gossip protocol.

The IDV client and the IDV server interact through the IDV protocol. The IDV protocol is an application protocol based on TLS, and the message format is shown in Fig. 4.

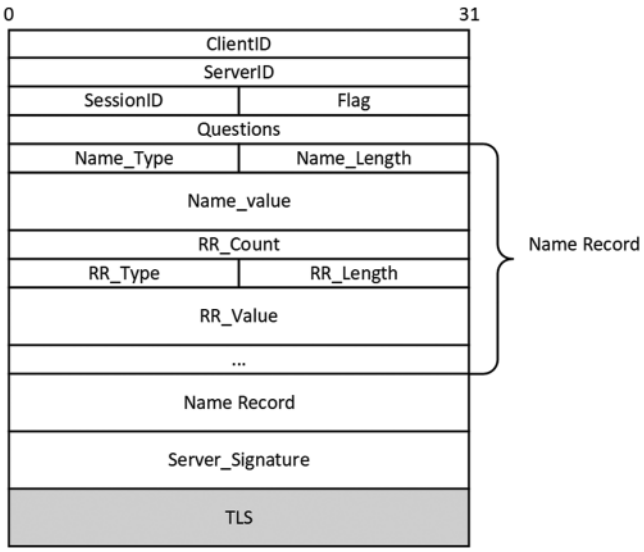


Figure 4: The message format of IDV

In Fig. 4, the field '*SessionID*' represents the unique number of this verification request, and the same value is used for the verification request and the verification response. The field 'Flag' is composed of 'T', 'OC', 'CR' and 'reserved', as shown in Fig. 5. Among them, '*T*' is the query/response flag, 0 is the query, and 1 is the response. '*OC*' is the operation flag, 0 means standard query, 1 means reverse query, '*RC*' is the return result, 0 means verification succeeded, and 1 means verification failed.

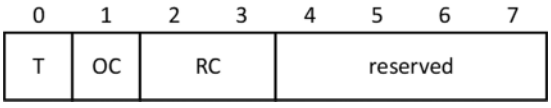


Figure 5: The composition of the flag

The field 'Questions' indicates the number of domain names requested for verification. Each domain name uses '*Name_Type*' as the domain identifier, and '*Name_Length*' and '*Name_Value*' indicate the length and value of the domain name, respectively. '*RR_Count*' identifies the number of resource records of the domain name, and '*RR_Type*', '*RR_Length*' and '*RR_Value*' identify the type, length and value of the resource record, respectively. The value of the resource record type is consistent with RFC1035. In the response message, if the value of the resource record is valid, keep the original value; otherwise, the value of '*RR_Length*' is 0, and the value of '*RR_Value*' is NULL.

4 Experimental Evaluation

IDV is a security-enhancing application for DNS. Compared with DNSSEC, IDV is a lightweight security solution. Unlike DNSSEC, IDV can provide security protection for domain names at all levels and has a good ability to gradually deploy. To verify the effectiveness of IDV and the ability to gradually deploy, we constructed two experiments. The first one is used to verify the effectiveness of IDV. The second one is used to verify the incremental deployment capability of IDV.

4.1 Experiment Setup

To evaluate the effect of our solution, we set up an experimental environment on the Internet. In this environment, we used virtual private servers (VPSs) to simulate different CDN servers, of which 2 VPSs are in Hong Kong (HK_VPS), 2 VPSs are located in Beijing (BJ_VPS), and 2 VPSs are located in Guangzhou (GZ_VPS). Two VPSs are in the United States (US_VPS). We deployed the Hyperledger Fabric platform and validation server on each server. The topology of our experiment is shown in Fig. 6. In this experimental environment, there is a malicious server that attempts to launch a DNS cache poisoning attack on the target host. We use Windows 7 as the operating system of the target host and Ubuntu as the operating system of the VPS.

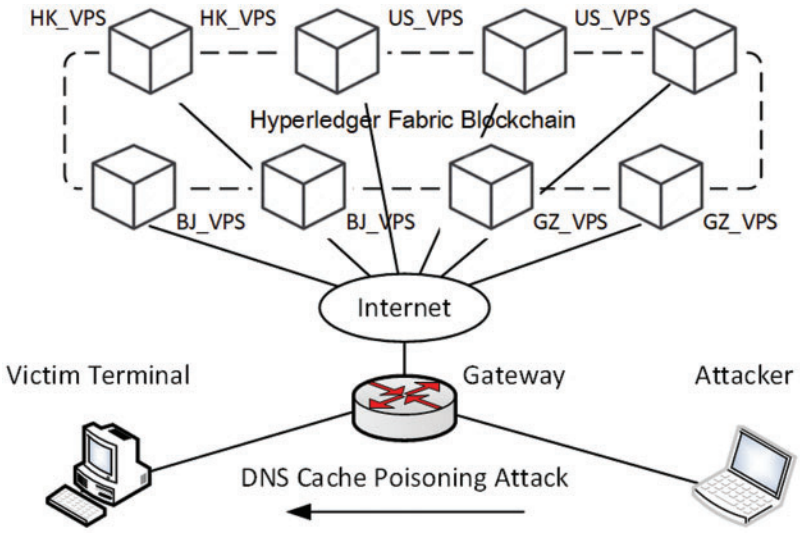


Figure 6: The environment of the experiment

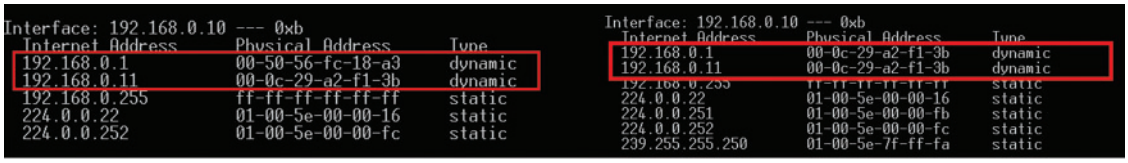
4.2 DNS Cache Poisoning Attack Resistance

In this experiment, we simulated a DNS cache poisoning attack to test the effectiveness of IDV. The domain names www.gzhu.edu.cn and www.jd.com were used for testing. We use the nslookup command to query the IP address corresponding to the domain name. The result is shown in Fig. 7. The domain name www.gzhu.edu.cn does not use CDN, and its correct IP address is 58.205.213.52. The domain name www.jd.com uses CDN, and the IP address of the Guangzhou CDN server is 120.238.202.131.

The attacker first uses the Ettercap tool [41] to implement an ARP spoofing attack on the victim terminal. After being spoofed, the victim terminal will replace the MAC address of the

gateway with the MAC address of the attacker's host. After the attack is successful, the attacker can impersonate the gateway and receive all traffic from the victim terminal, including DNS resolution requests. Figs. 8a and 8b show the gateway MAC address used by the victim terminal before and after the ARP attack, respectively.

Figure 7: Domain name resolving result without the poisoning attack



(a) (b)

Figure 8: The MAC address of the gateway (a) The correct MAC of gateway (b) The hijacked MAC of gateway

After gateway hijacking is successful, the attacking host sniffs the DNS query message of the target host. If it finds a DNS message querying the IP address of the domain name www.gzhu.edu.cn, it will forge a DNS response message, the IP in the message. The address is for the hacker host. The attack process is shown in Fig. 9.

Fig. 10 shows the fake DNS response message.

After the target host receives the forged DNS response message, it will cache the result in the local DNS cache, as shown in Fig. 11.

As mentioned above, when the local DNS cache of the victim terminal changes, the domain name newly added by the IDV client is sent to the IDV server to request verification. The IDV server will not be attacked by the attacker's DNS cache buffer poisoning, and the IDV server will identify false and fake domain name resolution results. When the IDV client receives the response from the IDV server, an error will be generated. As shown in Fig. 12, the client successfully detects the wrong merge and returns the correct IP address. Please note that the domain name (www.jd.com) has been successfully verified, the domain name uses CDN, and the verification server and the target host are not in the same zone.

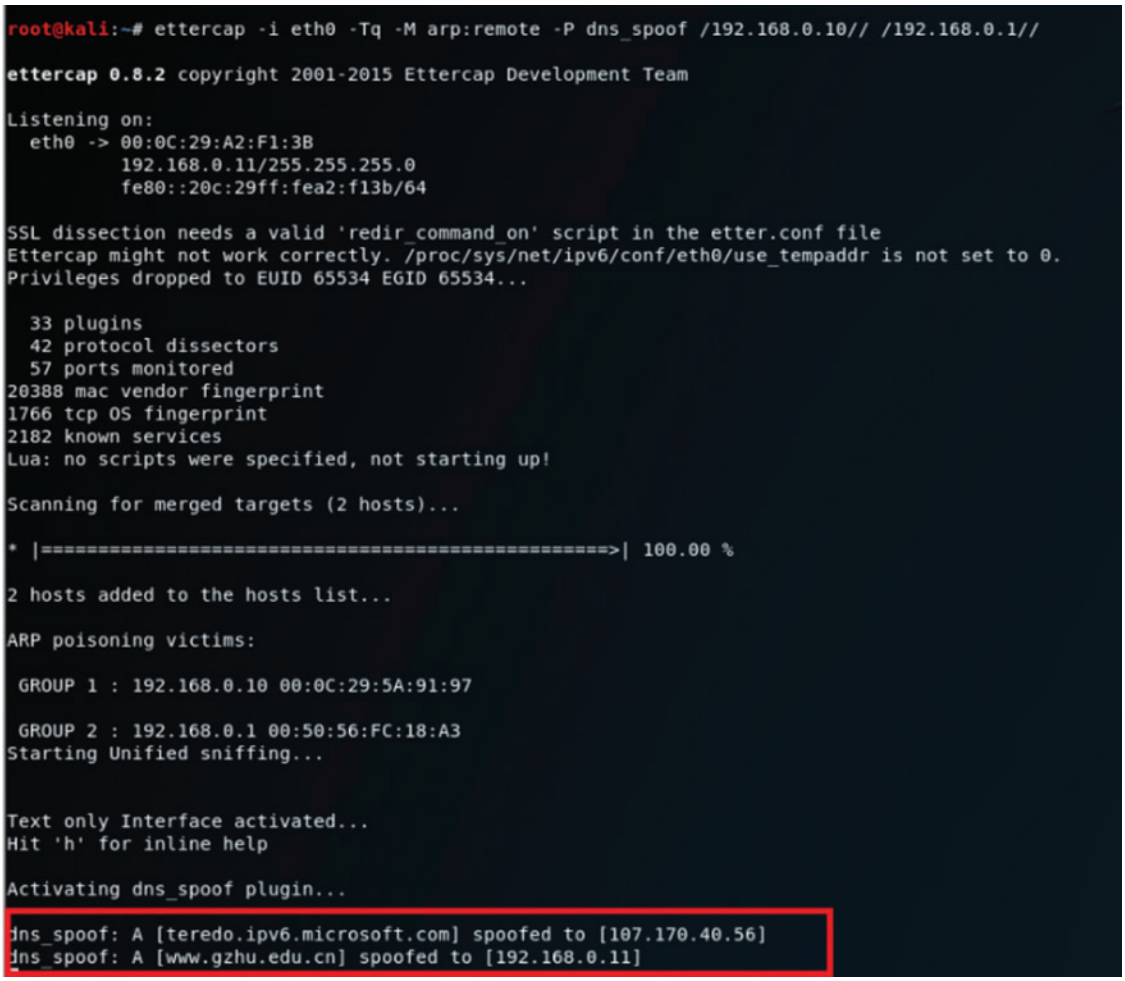


Figure 9: DNS spoofing attacks

In this experiment, we measure the time overhead caused by domain name validation. Some primary parameters are listed in Tab. 4.

In this experiment, time overhead mainly includes domain name verification processing overhead and network communication overhead. The processing overhead of domain name verification includes the processing time overhead of the client and server. The client's main processing content includes monitoring the local DNS cache, domain name verification message construction and sending time, and receiving and processing server response message time. The time cost of the communication part mainly includes the time cost of encryption processing and the time cost of consensus verification. The encryption processing time includes the secure communication time between the client and the server. This experiment tested the time cost of each sub-item to evaluate the time cost of the entire process.

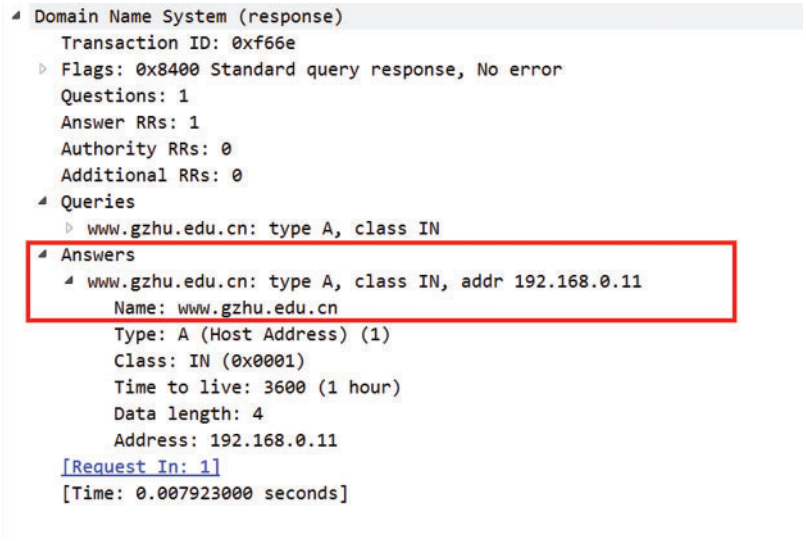


Figure 10: Forged DNS response packets

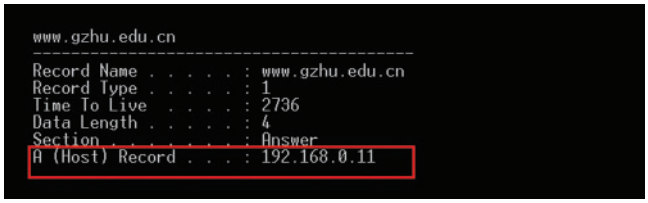


Figure 11: Target host DNS cache

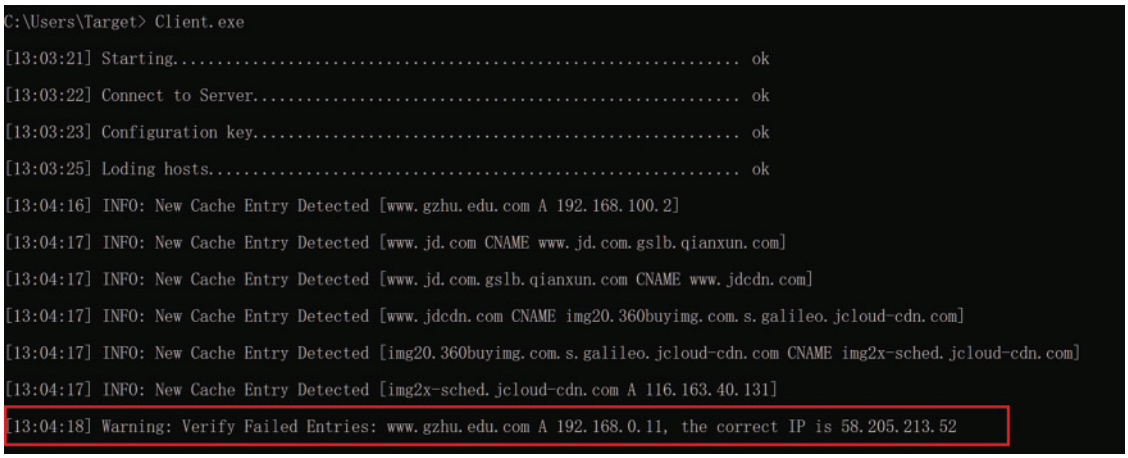


Figure 12: Validation alarm

Fig. 13 records the various time expenses during the experiment. The processing time overhead of the client (t1) and server (t2) is approximately 148 and 254 ms, respectively. The round-trip

delay (t3) between the client and the server is approximately 202 ms. The time overhead of local validation (t4) and consensus validation between CDN servers (t5) are approximately 205 and 288 ms, respectively. Without considering network performance, the main factor that determines the domain name verification delay is the efficiency of the consensus algorithm.

Table 4: Primary parameters for time overhead

Parameter name	Description
t1	Processing time of DNS client
t2	Processing time of CDN server
t3	Communication time between client and server
t4	Local validation time
t5	Consensus time between CDN servers

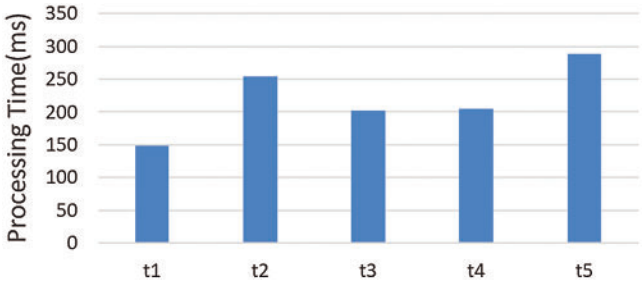


Figure 13: Sub-item statistics of time expenditure

To verify the efficiency of decentralized domain name verification, we used IDV verification and DNSSEC verification for the same domain name. Fig. 14 provides the time overhead in both cases. In Fig. 14, the processing latency of domain name verification through IDV is approximately 636 ms, while domain name verification through DNSSEC requires at least 1 s. The IDV method can complete the authenticity verification of the domain name within 1 s, and its verification efficiency is approximately 30% higher than that of DESSEC.

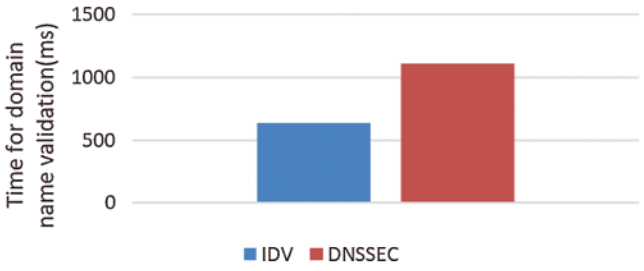


Figure 14: Comparison between IDV and DNSSEC

4.3 Monitoring Efficiency of Fake Resolving Result

IDV is a domain name authenticity verification solution that can be deployed gradually. It can be deployed gradually in a CDN network. As the number of deployed nodes increases, the coverage of global domain names will continue to increase. To verify the monitoring efficiency of IDV, we construct an experimental scene, as shown in Fig. 15. In Fig. 15, we gradually increase the number of IDV servers to observe the changes in domain name coverage.

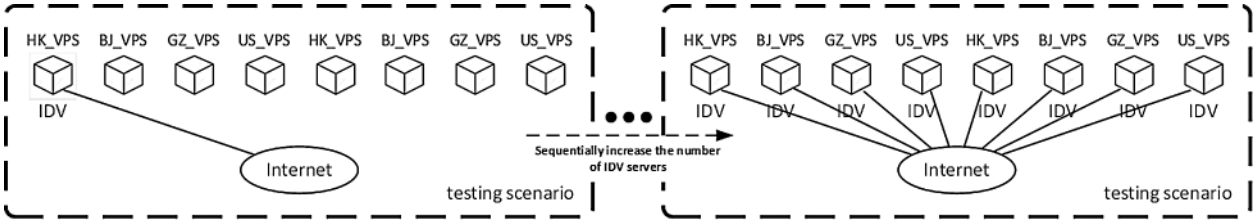


Figure 15: Monitoring efficiency experiment of IDV

To facilitate quantitative calculations, we define the following mathematical model. For ease of description, we assume that C_a represents the proportion of domain names whose authenticity can be verified by user terminal $\boldsymbol{\alpha}$, assume that X is the set of domain names that need to be tested, $|\boldsymbol{X}|$ represents the number of domain names in X, assume that Y is the set of servers deployed with IDV services, and $|\boldsymbol{C_\beta}|$ indicates the number of domain names that can be verified by server $\boldsymbol{\beta}$. From this, we can define the following formula:

$$C_a = \frac{\sum_{\beta \in \boldsymbol{Y}} |C_\beta|}{|\boldsymbol{X}|}, C_\alpha \epsilon [0, 1) \tag{1}$$

Formula (1) shows that when no IDV server is deployed, C_a is 0. As the number of IDV servers continues to increase, $\boldsymbol{C_a}$ will approach 1.

We collected DNS traffic data on the campus network of Guangzhou University from May 2020 to October 2020 and selected 10,000 domain names as the test set. To reduce the impact of invalid resolution on the experimental results, we removed those domain names that could not be resolved. We tested the 10 scenarios shown in Fig. 16 and counted the verifiable rate of domain names. The experimental results are shown in Fig. 16. Through Fig. 16, we find that when the number of IDV servers reaches 30% of the total number, the verifiable rate of the domain name is nearly 80%. This shows that only a general IDV server needs to be deployed in the CDN network, and a good effect of resisting DNS cache poisoning attacks can be achieved.

5 Discussion

It is necessary to discuss why IDV can be used to improve the security of Internet DNS. The following is an explanation from the four aspects of effectiveness, trustworthiness and incentives.

1. Effectiveness

DNS uses a hierarchical distributed structure. The domain name space is divided into different zones and stored in different authoritative servers, and no data synchronization is performed between these servers. This means that no DNS server can provide resolution and verification functions for all domain names. When the DNS server cannot provide domain name resolution,

it can only entrust other servers to perform domain name resolution in an iterative or recursive manner and then simply forward the resolution results to the requester. This is the root cause of attacks such as DNS cache poisoning, domain hijacking, and DNS spoofing.

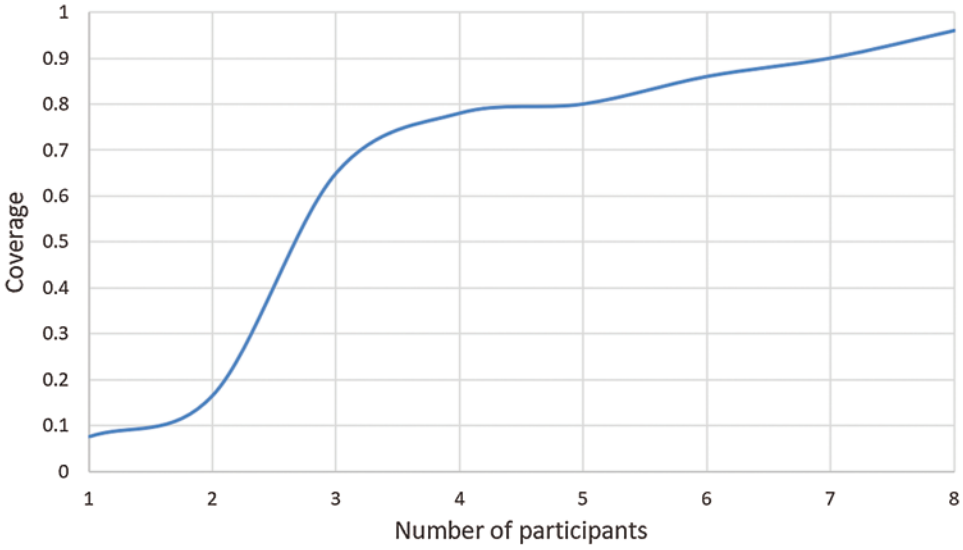


Figure 16: IDV coverage experiment result

DNSSEC realizes the authenticity verification of domain name resolution results by adding digital signatures to the resource records of domain names. Although DNSSEC based on PKI technology can theoretically guarantee the authenticity of domain names, it is very hard to deploy globally. The reason is very simple. We have no way to extravagantly ask domain name management agencies around the world to use a unified set of root certificates. In fact, the actual application effect of DNSSEC is not satisfactory. For example, Chung et al. notes that the protection rates of .com, .net and .org top-level domains (TLDs) are only 0.67%, 0.91% and 0.91%, respectively [42]. To make matters worse, instant DNSSEC can be deployed globally, but it still cannot completely prevent user terminals from being attacked by DNS cache poisoning. This is because the domain name verification process of DENSEC will not be initiated from the terminal. When there is a man-in-the-middle attack between the terminal and the recursive server, the terminal will not be able to verify the authenticity of the received domain name resolution results.

As a supplement to DNSSEC, IDV can realize the monitoring and authenticity verification of domain name resolution results on the user terminal side. First, IDV exists as an independent application; it will not affect the DNS resolution process of the domain name, and there is no need to change the design and implementation of the DNS protocol. Therefore, IDV is a lightweight security solution and easier to deploy. Second, IDV uses the information view of the CDN cache server. For example, one web page request is likely to result in more than 50 DNS requests. This means that the CDN cache server saves many domain name resolution results. These cached domain names are likely to be used again soon. After IDV verifies the authenticity of these domain name resolution results on the CDN server, it can provide a fast authenticity verification function for many subsequent visits. Finally, the experimental results of Section 4.3 show that a relatively ideal domain name monitoring range can be obtained by deploying a small amount of

IDV, and different CDN vendors can independently deploy their own IDV services without the need for a unified management centre. This makes IDV easier to deploy.

2. Trustworthiness

Compared with DENSEC, IDV does not rely on PKI to achieve the credibility of domain name resolution results. IDV mainly guarantees the credibility of domain name verification results from the following aspects. First, the process of a DNS cache poisoning attack is difficult to achieve before IDV. In DNS cache poisoning attacks, the attacker needs to guess the TCP session parameters and timing of the terminal or recursive server. Then, the attacker can imitate the false domain name resolution result and send it to the terminal or recursive server. This type of attack method is not applicable to IDV. The conversation process between the IDV client and server is protected by TLS. Attackers cannot tamper with or obtain communication content, so they cannot forge false responses. Second, IDV confirms the authenticity of the domain name through the CDN cache server. The CDN cache server can complete the authenticity verification of the domain name through multiple channels, and the attacker cannot intercept the verification message and verification timing of the CDN cache server. Therefore, it is very difficult to deceive the CDN cache server. Finally, when the IDV server publishes the verification results of the domain name, it will reach agreement among multiple nodes through the consensus mechanism of the blockchain. Even if a single node is deceived, it will be detected during the consensus process.

3. Incentive

Compared with DNSSEC, IDV has better incentives. As we know, DNSSEC is a design of a clean state. To realize the signature and verification of DNS resource records, DNSSEC adds four types of resource records: RRSIG (Resource Record Signature) records, DNSKEY (DNS Public Key) records, DS (Delegation Signer) records, and NSEC (Next Secure) records. If DNSSEC needs to be deployed on global recursive servers, all existing DNSs need to be replaced, which may affect the existing investments of ISPs and cause economic losses. In addition, recalculating record signatures for many assigned domain names is also an arduous task. Therefore, deploying DNSSEC on a global scale is not attractive to ISPs and users.

The CDN-based IDV is different. As the number of Internet online users and the demand for video-on-demand applications continue to grow, the (CDN) market, which was worth 11.76 billion U.S. dollars in 2020, is expected to reach 49.61 billion U.S. dollars by 2026 [43]. To improve the surfing experience of users, ISPs widely deploy CDN services to attract more users. Deploying the IDV service on the CDN can speed up network access while reducing the risk of users suffering from DNS cache poisoning. IDV does not require operators to change the operation and management process of the existing DNS, and IDV does not need to be deployed on all CDN cache servers, which further reduces the deployment overhead of IDV. Therefore, IDV is a security enhancement solution that can be supported by ISPs.

6 Conclusion

How to ensure the authenticity of DNS domain name resolution results is the core of DNS security issues. Because traditional DNS security solutions do not consider the limitations of their own centralization features, most of them have low efficiency and cannot be deployed gradually. This article proposes a blockchain-based decentralized domain name verification method, IDV. IDV does not rely on a single authoritative server and can arbitrarily choose a CDN that provides domain name verification services to complete domain name verification. Since CDN servers can be deployed globally and the number can be expanded, IDV is more scalable. In addition, because

IDV does not affect the working mode of traditional DNS, CDN users who have deployed IDV services can obtain fast domain name verification services. CDN users who have not deployed the IDV service will not have any other inconveniences except that they cannot use the domain name verification service. The research work in this article initially verified the effectiveness of IDV. In future research, we will further cooperate with CDN providers to try to apply IDV to the actual network environment and further verify its effectiveness.

Acknowledgement: Thank the support of National Natural Science Fund of China.

Funding Statement: This work was supported in National Natural Science Foundation of China (Grant Nos. 61976064, U20B2046), National Defence Science and Technology Key Laboratory Fund 61421190306), Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2019), National Key research and Development Plan (Grant No. 2018YFB1800702).

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

References

1. Orman, H., Streak, P. (2018). Blockchain: The emperors New PKI? *IEEE Internet Computing, 22(2),* 23–28. DOI 10.1109/MIC.2018.022021659.
2. Karaarslan, E., Adiguzel, E. (2018). Blockchain based DNS and PKI solutions. *IEEE Communications Standards Magazine, 2(3),* 52–57. DOI 10.1109/MCOMSTD.2018.1800023.
3. Ali, F. S., Kupcu, A. (2020). Improving PKI, BGP, and DNS using blockchain: A systematic review. *arXiv preprint arXiv*: 2001.00747.
4. Ølnes, S., Ubacht, J., Janssen, M. (2017). Blockchain in government: Benefits and implications of distributed ledger technology for information sharing. *Government Information Quarterly, 34(3),* 355–364. DOI 10.1016/j.giq.2017.09.007.
5. Khan, M. A., Salah, K. (2018). Iot security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems, 82,* 395–411. DOI 10.1016/j.future.2017.11.022.
6. Mattos, D. M. F., Krief, F., Rueda, S. J. (2020). Blockchain and artificial intelligence for network security. *Annals of Telecommunications, 75(3–4),* 101–102. DOI 10.1007/s12243-020-00754-7.
7. Saad, M., Anwar, A., Ahmad, A., Alasmary, H., Yuksel, M. et al. (2019). Routechain: Towards blockchain-based secure and efficient BGP routing. *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 210–218. Seoul, Korea (South).
8. Efficientip (2019). Understanding the critical role of DNS in network security strategy. https://www.efficient-ip.com/resources/idc-dns-threat-report$-$2019/.
9. Zou, F., Zhang, S., Pei, B., Pan, L., Li, L. et al. (2016). Survey on domain name system security. *IEEE First International Conference on Data Science in Cyberspace (DSC)*, pp. 602–607. Changsha, China.
10. Yin, S., Teng, Y., Hu, N., Jia, X. (2020). Decentralization of DNS: Old problems and New challenges. *ACM International Conference Proceeding Series,* vol. 2020, pp. 335–341. Guangzhou, China. DOI 10.1145/3444370.3444594.
11. Hu, N., Yin, S., Su, S., Jia, X., Xiang, Q. (2020). Blockzone: A decentralized and trustworthy data plane for DNS. *Computers, Materials & Continua, 65(2),* 1531–1557. DOI 10.32604/cmc.2020.010949.
12. Wachs, M., Schanzenbach, M., Grothoff, C. (2014). A censorship-resistant, privacy-enhancing and fully decentralized name system. *Cryptology and Network Security, 2014,* 127–142. DOI 10.1007/978-3-319-12280-9.
13. Wang, X., Li, K., Li, H., Li, Y., Liang, Z. (2017). ConsortiumDNS: A distributed domain name service based on consortium chain. *2017 IEEE 19th International Conference on High Performance*

Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), vol. 2017, pp. 617–620. DOI 10.1109/HPCC-SmartCity-DSS.2017.83.
14. Kaminsky, D. (2008). Black Ops 2008: It's the end of the cache as we know it. *Black Hat USA, 2*.
15. Network Working Group (2004). RFC 3833: Threat analysis of the domain name system (DNS) status.
16. Herzberg, A., Shulman, H. (2012). Security of patched DNS. *Computer Security–ESORICS, 2012,* 271–288. DOI 10.1007/978-3-642-33167-1.
17. Alharbi, F., Chang, J., Zhou, Y., Qian, F., Qian, Z. et al. (2019). Collaborative client-side DNS cache poisoning attack. *IEEE INFOCOM, 2019,* 1153–1161. DOI 10.1109/INFOCOM.2019.8737514.
18. Thomdapu, S. T., Katiyar, P., Rajawat, K. (2021). Dynamic cache management in content delivery networks. *Computer Networks, 187107822,* 1–16. DOI 10.1016/j.comnet.2021.107822.
19. Palladino, N., Santaniello, M. (2021). *IANA functions, ICANN, and the DNS war*. Information Technology and Global Governance, British, Palgrave Macmillan.
20. Namecoin (2021). Against censorship. https://www.namecoin.org/.
21. Nakamoto, S. (2009). Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/en/bitcoin-paper.
22. Ali, M., Nelson, J., Shea, R., Freedman, M. J. (2016). Blockstack: A global naming and storage system secured by blockchains. *Proceedings of the 2016 USENIX Annual Technical Conference*, pp. 181–194. Denver, USA.
23. ENS (2021). Decentralised naming for wallets, websites, & more. https://ens.domains/.
24. John, G. (2021). Peername: Surf blockchain-based domains. https://peername.com/.
25. EMCDNS (2021). Emercoin blockchain. https://emercoin.com/en/.
26. He, G., Su, W., Gao, S., Yue, J. (2020). Td-root: A trustworthy decentralized DNS root management architecture based on permissioned blockchain. *Future Generation Computer Systems, 102,* 912–924. DOI 10.1016/j.future.2019.09.037.
27. Eyal, I., Sirer, E. G. (2014). Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM, 61(7),* 436–454. DOI 10.1145/3212998.
28. Simplified name verification protocol. https://blockstack.org/.
29. Madariaga, D., Madariaga, J., Panza, M., Bustos-Jiménez, J. (2021). Detecting anomalies at a TLD name server based on DNS traffic predictions. *IEEE Transactions on Network and Service Management, 18(1),* 1016–1030. DOI 10.1109/TNSM.4275028.
30. Lyu, M., Gharakheili, H. H., Russell, C., Sivaraman, V. (2021). Hierarchical anomaly-based detection of distributed DNS attacks on enterprise networks. *IEEE Transactions on Network and Service Management, 18(1),* 1031–1048. DOI 10.1109/TNSM.4275028.
31. Maksutov, A. A., Cherepanov, I. A., Alekseev, M. S. (2017). Detection and prevention of DNS spoofing attacks. *Siberian Symposium on Data Science and Engineering,* pp. 84–87. Novosibirsk, Russia. DOI 10.1109/SSDSE.2017.8071970.
32. Alibaba Cloud (2021). https://www.alibabacloud.com/.
33. Tencent Cloud (2021). https://intl.cloud.tencent.com/.
34. Akamai (2021). https://www.akamai.com/uk/en/.
35. Cloudflare (2021). The cloudflare global anycast network. https://www.cloudflare.com/network/.
36. Aws (2021). https://aws.amazon.com/.
37. Google Cloud (2021). https://cloud.google.com/.
38. Azure (2021). https://docs.microsoft.com/en-us/azure/cdn/cdn-pop-locations.
39. Man, K., Qian, Z., Wang, Z., Zheng, X., Huang, Y. et al. (2020). DNS cache poisoning attack reloaded: Revolutions with side channels. *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1337–1350. Virtual Event, USA. DOI 10.1145/3372297.3417280.
40. Hyperledger (2021). https://www.hyperledger.org/.

41. Ettercap (2021). https://www.ettercap-project.org/.
42. Chung, T., Rijswijk-Deij, R. V., Choffnes, D., Maggs, B. M., Mislove, A. et al. (2017). A longitudinal, end-to-end view of the DNSSEC ecosystem. *Proceedings of the 26th USENIX Security Symposium*, pp. 1307–1322. Vancouver, Canada. DOI 10.5555/3241189.3241291.
43. Gartner (2021). https://www.gartner.com/reviews/market/global-cdn.