IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A Trustworthy Block chain-based Domain Name System

Dr.P.M.S.S.Chandu[1],M.E, Ph.D M Jashwanth[2] R Gayathri[3] K JayaKiran[4] G Bhuvanesh[5].

[1]Professor, [2345]UG Student [12345]Department of Computer Science and Engineering,
[12345]Siddharth Institute Of Engineering & Technology, Tirupathi, Andhra Pradesh, India.

ABSTRACT

T 2 DNS is a third-party DNS service that we created. T 2 DNS provides client trustworthiness evidence, protects clients from channel and server attackers, is compatible with the current Internet infrastructure, and introduces bounded overhead while serving client DNS queries. T 2 DNS uses an encryption and obfuscation hybrid protocol to protect user privacy, and Intel SGX is used to power its service proxy. Scaling the initialization procedure, limiting the obfuscation overhead, and fine-tuning useful system parameters are obstacles we successfully overcame. We develop a T 2 DNS prototype, and testing demonstrates that it is completely functional, has a manageable overhead when compared to alternative alternatives, and scales to the number of customers.

Index Terms: *Ethereum, smart contracts, DNS security, block chains.*

INTRODUCTION

Without a question, the internet has been crucial in the twenty-first century. In the spring of 2020[9], the corona virus (COVID-19) outbreak, 34% of Austrians and 53% of Americans reported using the Internet more frequently than normal. Seventy percent of respondents to a global survey of CIOs conducted in March indicated that they presently work from home. Furthermore, about 30% of respondents said they anticipated working remotely permanently. Using the Internet, workers can work from home, and students in an epidemic region can learn remotely. Cybercriminals employ a range of methods, such as phishing, malware, denial of service, and others, to initiate cyberattacks. Phishing is the most prevalent of these assaults. where criminals pretend to be trustworthy websites in order to deceive users into giving up personal information or data, such as usernames, passwords, and credit card numbers. states that a new phishing website appears every 20 seconds and claims that 74% of phishing websites use the HTTPs protocol to deliver their content. VeriSign mistakenly granted two digital certificates to a person posing as a Microsoft representative, the company alerted customers, allowing the fraudster the means to lure them into a website hosting harmful software. Visitors to a domain that the attacker had compromised were reported using a novel technique[8]. A "security certificate update" and two malware versions are downloaded as the victims are prompted to do so in order to steal their PC's data.

federated block chain, also known as consortium block chain, is a type of block chain technology that only certain organisations or groups can use and that requires members to register [18]. The Consortium block chain is password-protected. This indicates that only the consortium members who are actively participating have read and write rights. Two nodes make up the consortium block chain; one node maintains communication with clients and other nodes, while the second node manages private transactions by carrying out cryptographic operations. A permissioned network does not require proof of work (PoW), hence consortium block chains offer a variety of consensus procedures. The consortium must give its approval before a transaction can be verified or a smart contract can be issued, although the general public is free to consult and transact. As a result, only authorised nodes and users can access any services provided by the consortium block chain. the execution of transactions and smart contracts can be done by either permissioned users or other users.

LITERATURE SURVEY

Describe a network. "Any group of interlinking lines resembling a net, a network of roadways or an integrated system, a network of alliances" is the definition of "network." Simply described, a computer network is a collection of linked computers.

P2P: A peer-to-peer network is one in which each computer serves as both a client and a server and lacks a dedicated server. When there are ten or fewer users nearby and they are close to one another, this is a viable networking option. Because users will be the ones granting rights for shared resources, a peer-to-peer network can be a security nightmare.

CLIENT/SERVER: This form of network uses a dedicated server or servers to service a large number of users. In order to access the server(s) and run programmes or download files, clients must log in. One or more administrators can control security and permissions, which prevents the aforementioned computer novices from tampering with things they shouldn't be. Also, this kind of network enables convenient backup services, lowers network traffic, and offers a wide range of other services that are included with the network operating system (NOS).

CENTRALIZED: This client/server model, which is most frequently used in UNIX environments, also uses "dumb terminals" as clients. The client might not have a floppy drive as a result of this All applications and processing are done on the server(s), whether it be a hard drive or CDROM. As you can expect, this calls for incredibly expensive and quick server(s). On this type of network, security is extremely high, albeit a similar level of security can be attained by using an NT server with the proper settings.

Describe a DNS server: The phonebook of the Internet is the Domain Name System (DNS). DNS determines the right IP address for websites when users enter domain names like "google.com" or "nytimes.com" into web browsers[6]. When communicating with origin servers or CDN edge servers to obtain website content, browsers use such addresses. All of this is made possible by DNS servers, which are computers that only deal with DNS inquiries.

Describe a server. A server is a tool or software created specifically to offer services to other programmes, sometimes known as "clients". The majority of contemporary desktop and mobile operating systems include DNS clients that let web browsers communicate with DNS servers. The Client-Server Model has more information.

In what way do DNS servers answer a DNS request?

Describe a server. A server is a tool or software created specifically to offer services to other programmes, sometimes known as "clients". The majority of contemporary desktop and mobile operating systems include DNS clients that let web browsers communicate with DNS servers. The Client-Server Model has more information.

In what way do DNS servers answer a DNS request?

Recursive resolvers, root name servers, TLD name servers, and authoritative name servers are the four servers that collaborate to send an IP address to the client in a typical DNS query without any caching. The DNS recursor, also known as the DNS resolver, is a server that accepts the DNS client's query and then communicates with other DNS servers to find the right answer.

The resolver really acts as a client after receiving the client's request and queries the other three categories of DNS servers to find the correct IP.

The resolver first makes a query to the root name server. The root server is the initial place where human-readable domain names are converted (resolved) into IP addresses. The address of a TLD DNS server (such as.com or.net), which holds the data for its domains, is then returned to the resolver by the root server. The resolver then makes a query to the TLD server. The IP address of the domain's official name server is returned by the TLD server. The authoritative name server will respond with the IP address of the origin server when the recursor has contacted it.

Finally, the resolver will give the client the IP address of the origin server. The client can then submit a query to the origin server using this IP address, and the origin server will react by returning website data that the web browser can understand and display.

Describe DNS caching.

Recursive resolvers can use cached data to resolve DNS queries in addition to the method described above. The resolver will then save that data in its cache for a finite period of time after obtaining the proper IP address for a specific website. If any other clients send requests for that domain name during this time, the resolver can forego the usual DNS lookup procedure. and merely reply to the client with the cached IP address. The resolver must obtain the IP address once again after the caching time limit has passed in order to add a fresh entry to its cache. The DNS records for each site expressly define this time restriction, often known as the time-to-live (TTL). The TTL typically ranges from 24 to 48 hours. Web servers occasionally change their IP addresses, thus resolvers can't continuously serve the same IP from the cache, necessitating a TTL.

What occurs when DNS servers malfunction?

Many factors, including lost power, cyberattacks, and technical issues, can cause DNS servers to malfunction. When the Internet first began, DNS server disruptions may have a major impact. Fortunately, DNS now has a lot of redundancy built in. For instance, there are numerous instances of the root DNS servers and TLD name servers, and the majority of ISPs provide their customers with backup recursive resolvers. (Single users can also use publicly accessible DNS resolvers, such as 1.1.1.1 from Cloud flare.) The majority of well-known websites also use numerous instances of their reliable name servers.

Some users may experience delays in the event of a severe DNS server failure as a result of the volume of requests being handled by backup servers, but it would require a DNS outage of extremely massive proportions to render a sizable chunk of the Internet unreachable. This actually occurred in 2016, when one of the largest DDoS attacks in history hit DNS service Dyn. With built-in DNS security, Cloud Flare's Managed DNS Service offers protection for DNS servers[4] against assaults and other typical causes of server failure.

DNS filtering

Blacklists are used in our DNS filtering to ban specific websites or IP addresses that are known to be dangerous. Yet, DNS filtering is simple to get around by using an alternative DNS or adding entries to the host server. With the rapid advancement of the new block chain technology, we adapted to make use of its advantages to offer a safe DNS mechanism.

RECENT WORKS

Many studies have also attempted to use its characteristics to offer a secure DNS method. Hari et al. have suggested the first piece of work. who created the DNS infrastructure primarily rely on PKI. A system called distributed decentralised domain name service (DNS), which is built on a distributed hash table and makes use of a domain name ownership scheme, was proposed by Benshoof et al. [3]. also made use of decentralisation to propose a DNS resolution technique that reduces single points of failure and data tampering related to domain name resolution. A DNS Cache Resources Trusted Sharing Model, sometimes known as DNS[5,] was additionally proposed. They asserted that the model can increase the validity of DNS resolution outcomes. In the meanwhile, they offered a stochastic distributed decentralised storage technique to address the consortium's low efficiency issue.

LIMITATIONS

Is really sluggish and uses a lot of power. This is complicated and unable to build an infrastructure that completely protects privacy.

does not adhere to security restrictions.

PROPOSED METHOD

The theoretical architecture of the system. Every industry and corporation assembles to build its own consortium block chain. Nobody can change their unified resource locator (URL) or Internet protocol (IP) because they uploaded them to blocks in a chain. A new person or organisation must first be looked into and approved by the consortium before being allowed to join the consortium block chain. They are permitted to join the consortium and initiate a smart contract to conduct a transaction that uploads their URL and IP to a block if their personal/organizational data is accurate, the content of their website and URL is correct, and they have coherent URLs. Authentication cannot be passed by a malicious attacker due to a lack of actual legitimate business data.

To ensure the reliability of the user's domain name resolution results, the threat will be segregated from the trusted network.

Using Block Chain Technology, we protect the DNS Entries (i.e., the domain name and IP address bindings) in the DNS Server.

CONTRIBUTIONS

Chains of blocks that nobody can alter arbitrarily. Their individual or organisational data is verified. Because of a lack of verifiable formal business data, malicious attackers cannot pass the authentication. To confirm the validity of the user's domain name resolution results, the threat will be isolated.

ALGORITHM:

INPUTS:

URL	-	Uniform Resource Locator
IPaddr	-	IP Address
ReserveYears	-	no of years registration life
Sig	-	Signature
PKey	-	Private Key

OUTPUTS:

Result	-	IP address of the domain

```
Insert(url, ip, ry, sig, pk, o):
    created = false
    IF exists url in URL OR ip in IP Then
        return created
    Else:
        hashurl = hash256(url)
        haship = hash256(ip)
        msg = encode.Packrd(hashurl, hashIP)
        IF Eth.Decrypt(pk, sig)<> msg;
            return created
        Else:
            ttl = getTTLValue(reserveYears)
            created=createNewDNSEntry(url, ip, ttl, o)
            return created
        End IF
    End IF
End
Query(url):
    IF not exists url in URL:
        result.push("404 Not Found")
    ELSE
        result.push(IP[url])
        return result
    EndIF
END
```

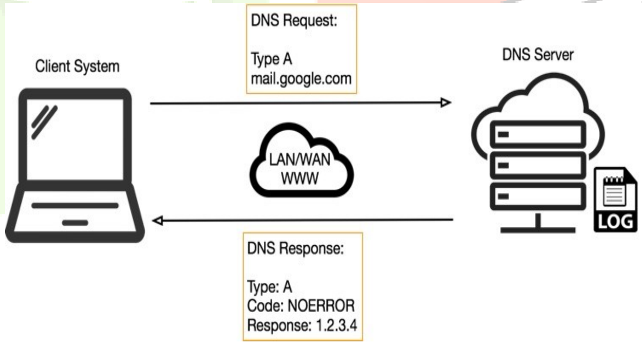

Fig:1 Architecture

FUNCTIONAL MODULES:

DNSSecurity Module

Keep an eye out for domain name system (DNS) registry data that has been questioned that could compromise external DNS servers that can be utilised for targeting. This activity will be largely hidden from the target organization's view, making it challenging to identify this conduct. Detection efforts could concentrate on associated phases of the adversary lifecycle, such Command and Control.

Keep an eye out for logged domain name system (DNS) registry information that could compromise external DNS servers that could be targeted. This activity will be largely hidden from the target organization's view, making it challenging to identify this conduct. Detection efforts could concentrate on associated phases of the adversary lifecycle, such Command and Control.

DNS Server Module

The phonebook of the Internet is the Domain Name System (DNS). DNS is in charge of determining the correct IP address for websites when users enter domain names like "google.com" or "nytimes.com" into web browsers. When communicating with origin servers or CDN edge servers to obtain website content, browsers use such addresses. All of this is made possible by DNS servers, which are computers that only deal with DNS inquiries.

Module for domain name query

All consortiums will build the top-level domain (TLD) block chain to enhance query performance. In a DApp, a user types a URL to communicate with a smart contract that has been set up on the block. After that, a DNS request will be made to the TLD block chain by the smart contract. The user's actions have

Module for domain name resolution

The root block chain is constructed by, including Internet corporations for assignment, as the overall process of this phase.

A smart contract responding to a user's request will send the initial DNS query to the root block chain. The address of the smart contract in the target url block chain is returned by the root block chain. Then, a search for the location of the authoritative block chain is automatically started by the target smart contract in the url block chain.

RESULTS

Findings The proposed mechanism combines the consortium blockchain's consensus process, key management, and participant authentication to deliver a more secure, dependable, and trusted DNS resolving service. We examine the proposed program's security in this Section. We contrast our plan with earlier, well-known designs, and table 4 summarises the comparison of the key elements. The stratification functionality mimics the way that recursive DNS queries are distributed hierarchically to various blockchain-based domains in the present global domain name system. This feature prevents large queries from being sent to recursive servers, which puts more strain on the network's resources and creates a service bottleneck. The comparative result demonstrates that the proposed mechanism has superior security than others in addition to the stratification feature. The following are the main benefits: 1) Trustworthy domain name services The integrity and dependability of data on the chain can be ensured by the blockchain. But, if a malevolent applicant uploads the data, the phoney In our system, the applicant is checked against the application's legal documentation before the domain name is registered. The consortium's consensus controls the on-chain data. The block structure includes the hash of the previous block and a timestamp (prehash). All on-chain data must be modified in order to change any block, which is not possible. The new block will contain a recent timestamp that is simple to check if the applicant registers a new domain name with the same IP address.

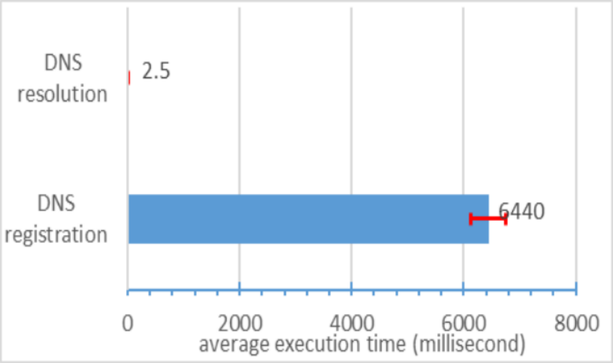

Fig:2 Comparison graph

The suggested DNS can provide dependable service as a result of the chain's information being shared in an open and secure manner. 2) Be confident in key-based authentication Only authenticated nodes are permitted to participate in message exchange on a permissioned blockchain. Sender nodes are verified using the key-based authentication technique using a recoverable signature. Each user is given a set of asymmetric keys produced on the elliptical curve. Exchanged messages are signed by each sender using recoverable, which enables the recipient to extract the sender's public key from the message signature. The recipient contrasts the extracted public key with the list of public keys belonging to other permissioned nodes. Only if one of the items matches does the receiver confirm the sender node's authenticity. Otherwise, the link is turned down by the receiver. To ensure that no single node can make a crucial modification, Quorum requires a predetermined number of authenticators. Moreover, actions in Quorum demand a large enough number of nodes to provide their credentials.

CONCLUSION AND FUTURE WORK

In this study, we offer a novel and secure domain name service based on consortium block chain, after referencing earlier works of block chain-based DNS solutions. A reliable DNS registration and resolution method using smart contracts is suggested. In order to achieve a good performance while guaranteeing the security of domain name resolving, the given mechanism makes use of the Quorum with Raft consensus algorithm. An extensive examination and comparison of the performance results between Raft and IBFT are presented. We think that the suggested DNS technique can successfully fend off various assaults. In the future, we'll work to efficiently integrate the suggested technique with the current DNS. In order to create a more effective and secure system, we will also construct a certificate authority that enhances Quorum's transaction authentication and remote user authentication.

REFERENCES

[1] wandera.com, “Mobile Threat Landscape Report,” 2020. [Online].:

[2] apwg.org, “Q4 2019 Phishing Activity Trends Report,” 2021. [Online]. Available: https://docs.apwg.org/reports/ apwg trends report q1 2021.pdf.

[3] B. Fonseca, “VeriSign issues false Microsoft digital certificates,” ComputerWorld:Security, Mar. 23, 2001. [Online]. Available: https://www.computerworld.com/ article/2798454/verisign-issues-falsemicrosoft-digital-certificates.html.

[4] C. Osborne, “Backdoor malware is being spread through fake security certificate alerts,” ZDNet: Security, Mar. 5, 2020. [Online]. Available: https://www.zdnet.com/article/backdoor-malware-is-beingspread-through-fake-security-certificate-alerts/.

[5] C. Cimpanu, “Let’s Encrypt to revoke 3 million certificates on March 4 due to software bug,” ZDNet: Security, Mar. 4, 2020.

[6] A. Har and T. V. Lakshman, “The Internet blockchain: A distributed, tamper-resistant transaction framework for the Internet,” in Proc. ACM HotNets, 2016.

[7] B. Benshoof, A. Rosen, A. G. Bourgeois, and R. W. Harriso, “Distributed decentralized domain name service,” in Proc. IEEE IPDPSW, May 2016.

[8] J. Liu, B. Li, L. Chen, M. Hou, F. Xiang, and P. Wang, “A data storage method based on blockchain for decentralization DNS,” in Proc. IEEE DSC, Jun. 2018.

[9] Z. Yu, D. Xue, J. Fan and C. Guo, “DNSTSM: DNS cache resources trusted sharing model based on consortium Blockchain,” IEEE Access, vol. 8, pp. 13640-13650, 2020.

[10] C. Cachin and A. Samar, “Secure distributed DNS,” in Proc. IEEE/ISIP ICDSN, 2004.