Distributed Domain Name System Security Solution

Mukesh Kumar Bansal[1], M. Sethumadhavan[1], Venkataraman Sarma[1], M. K. Gupta[2*]

[1]TIFAC-CORE in Cyber Security, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India

[2]Department of Electrical Engineering, Suresh Gyan Vihar University, Rajasthan, India.

Abstract: - In view of the vulnerability of Domain Name System DNS) to multiple type attacks posing security threats, a disruptive protocol to ensure DNS security is proposed in this paper. The use of root server by DNS system which is owned by agencies other than user, there is ample chance of cyber threats viz. cache-poisoning, denial of service and other types of inceptions. The available solutions for overcoming such security threats are limited by its' serviceability. The present paper reviews the concerns for the cyber threat of DNS and analyses the criticalities involved in existing solutions towards DNS security. Introducing the concept of block chain technology, a proposition for a Block chain-based solution named 'DNS-B Chain' as an intermediate solution, which may bring in a philosophical change in DNS security solutions. The contemplated development uses Hyperledger Fabric where DNS client is created using Java and ledger entries are stored in CouchDB to maintain DNS Cache; querying is accomplished by the use of Hyperledger APIs. Thus, secure and immutable solutions It is configured to solve issues like DNS Cache poisoning or corrupting DNS data and censorship / restrictions on DNS servers. Finally, discussion is made on the future improvements of the technology, with the advocacy of how block chains may become a vital resource in trusted computing and security with inbuilt DNS Security.

Keywords— DNS, DNSSEC, DNSCURVE, Blockchain, Hyperledger)

1. Introduction

The Domain Name System (DNS) is the phonebook of the Internet. Users access information online through do-main names, like espn.com. However, Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so that browsers can load Internet resources. Each device connected to the Internet has a unique IP address which other machines use to find the said device [1]. DNS servers eliminate the need for users to memorize IP addresses such as 192.168.1.1 (in IPv4), or more complex newer alphanumeric IP addresses such as 2400:cb00:2048:1::c629:d7a2 (in IPv6). The process of DNS resolution involves converting a hostname (such as www.example.com) into a computer-friendly IP address (such as 192.168.1.1). The IP address given to each device on the iinternet is necessary to find the appropriate Internet device - like a street address is used to find a home. When a user wants to load a webpage, a translation must occur between what a user types into their web browser (example.com) and the machine-friendly address necessary to locate the example.com web page [2,3].

2. Review of issues involving DNS security-B Chain

2.1 Blockchain

A blockchain is a type of ledger which increases regularly. It maintains a stable proof of all the transactions. Through this all the transactions are carried out in a safe, sequential, and incontrovertible manner **[1-4] As** name suggests, it is considered as chain of blocks in which information are stored. Once a transaction takes place it is automatically stored by each block and goes into the blockchain in the form a stable record. As soon as a block fulfilled, another new block is generated as shown in figure 1.

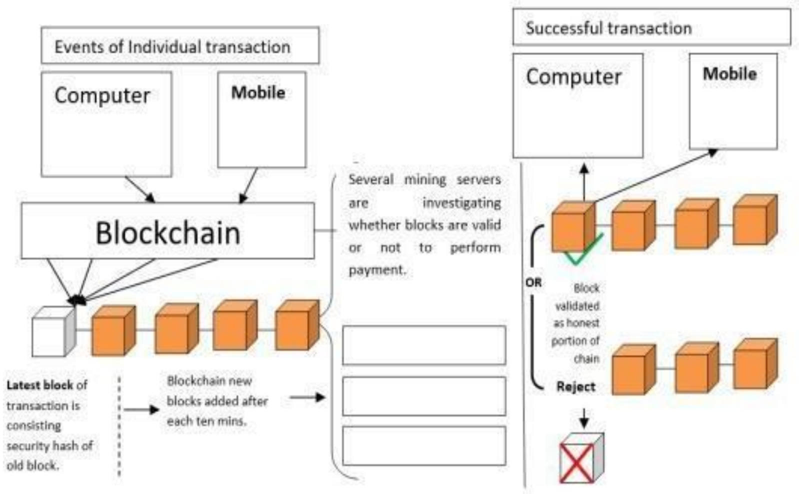
Events of Individual transaction
Computer
Mobile
Blockchain
Several mining servers are investigating whether blocks are valid or not to perform payment.
Latest block of transaction is consisting security hash of old block.
Blockchain new blocks added after each ten mins.
Successful transaction
Computer
Mobile
OR
Block validated as honest portion of chain
Reject

Figure 1: Blockchain Overview

2.1.1 Advantages of blockchain

This technology provides a lot of benefits. One of the most important advantages of block chain is that it is openly available for use. All users have power to observe the blocks as well as transactions that is collected. For protection of all transaction's details, a secret key is used. It is formed in such a way that it can be used in the form of private Blockchain where nodes are inserted in chain once the permission is obtained. In blockchain based solution, an individual cannot give the permission of transactions [5-7]. Similarly, not an individual unit can put definite policy related to recognition of transactions. Above two points' shows that the system is quite safe. For approval of transactions, participants that is the nodes in blockchain are required. Since security is a very important thing, expansion of records is allowed but no one is allowed to make alteration in previous report. Few other advantages of blockchain are:

- Ledger: It is type of document which increases regularly.
- Stable: It signifies that as soon as a transaction takes place it can accommodate it permanently in the ledger.
- Safe: Records have been in a very safe manner. It applied high level coding to lock information in blockchain.
- Sequential: It indicates that each transaction takes place after the previous one.
- Incontrovertible: It signifies once you build all the trans-action onto the blockchain, this ledger can never be misused.

2.2 Procedure of Blockchain

As soon as fresh trans-action is inserted, all participants give their permission for transactions. Transaction is verified by the applicationof an algorithm (smart contract). On the basis of Blockchain system accurate numbers of participants which are requiredfor the validation of transaction are defined [9]. It can vary depending upon the systems. For the successful validation of transaction it is necessary that the majority of the partic- ipants give their permission. In the following step, group ofcertified transactions is established in block. It is allocated to all the nodes present in the network. A fresh block is certified by each and every participant. Every consecutive block contains a hash [10]. It is a digital impression of the present block. This system becomes safe in the company of shared computing system. It gives high Byzantine fault tolerance.

2.2.1 Challenges

Along with advantages of Blockchain model, many limitation and problems are to be considered. Challenges have been explained below [11].

- Limited Knowledge – Although a lot of deliberations have been there, the understanding about the importance of block chain is yet to be adequate. In addition, people do not know the manner in which this technology has been put in to operation in different situations
- Lack of developers – Unlike many other field of technology, the number of specialized technology developers are scarce in the field of block chain
- Adaptability - Blockchain is considered very new technology without many case studies in industry. Hence decision makers are hesitant in focusing on using blockchain in solving their industry issues

2.3 DNS

At its fundamental core, the DNS is a list of names and their equivalent IP addresses, but the methods for creating, storing, and retrieving those names are very different from those in a host table. DNS consists of three elements [2-6]:

2.3.1 Namespace:

DNS has a hierarchical inverted tree structure, which is called DNS namespace. Each branch of DNS namespace tree is a domain, each subbranch is a subdomain. Each domain contains a collection of resource records that contain host names, IP addresses, and other information. This DNS namespace tree is queried to retrieve specific resource records from a particular domain

2.3.2 Recursive server

The first DNS query from client interacts with the Recursive Server, which is the server that responds to a recursive request from a client and takes the time to track down the DNS record. It does this by making a series of requests until it reaches the authoritative DNS nameserver for the requested record (or times out or returns an error if no record is found).

2.3.3 Authoritative Server (Name Servers)

These are the Internet's equivalent of a phone book. They maintain a directory of domain tree structure and translate them to IP addresses. They are capable of responding to queries for information about the domains for which it is the authority and also of forwarding queries about other domains to other name servers. This enables any DNS server to access information about any domain in the tree. It provides original and definitive answers to DNS queries. It does not provide just cached answers that were obtained from another name server. Therefore, it only returns answers to queries about domain names that are installed in its configuration system. There are two types of Authoritative Name Servers [12-14]:

2.3.3.1 Master server (primary name server)

A master server stores the original master copies of all zone records. A hostmaster only make changes to master server zone records. Each slave server gets updates via special automatic updating mechanism of the DNS protocol. All slave servers maintain an identical copy of the master record.

2.3.3.2 Slave server (secondary name server)

A slave server is exact replica of master server. It is used to share DNS server load and to improve DNS zone availability in case master server fails. It is reciommended that there are at least have 2 slave servers and one master server for each domain name. The multiple name servers make sure that the domain is still functional even if one name server becomes inaccessible or inoperable due to security or overloading issues. On the internet each domain name assigned a set of authoritative name servers. You can find out authoritative name servers by typing the following command at shell prompt **host-t ns dnsknowledge.com.**

Thus, the authoritative name server and the recursive name server play the major role in DNS resolution. The authoritative server and the recursive server are registered with the DNS using unique identification. This enables to identify the genuine servers [15].

2.3.4 DNS Message Format: This section describes about basic DNS message format used for all DNS operations (queries, responses, zone transfers, notifications, and dynamic updates). The basic DNS message begins with a fixed 12-byte header followed by four variable-length sections

- Questions (or queries)
- answers
- Authority records
- Additional records

2.3.5 DNS Communication

DNS communication to resolve domain name query between client and DNS server [2,3] is explained in this section. DNS Query resolution takes a fraction of second. Comunications steps are: i) client requests recursive server for Website Information. First, user visits a website by typing a domain name into a web browser, ii) recursive servers connect with root server to get info on authoritative server, iii) Query is raised to the Authoritative DNS Servers, iv) accessing secondary authoritative server and the DNS record the information is finally passed to recursive server and client. The Fig 2 shows the various steps used as part of DNS communication from client to DNS Servers [16, 17].

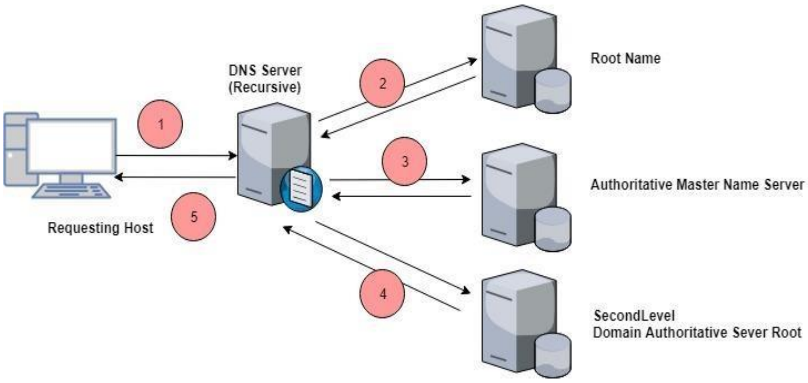
DNS Server
(Recursive)
Root Name
1
2
3
4
5
Requesting Host
Authoritative Master Name Server
SecondLevel
Domain Authoritative Sever Root

Figure 2: DNS Communication

3. DNS Security Issue and Solution

In this section, a few of DNS threats and available solutions (Protocol level solutions as well as Block Chain based solutions) are discussed.

3.1 DNS security threats

The current DNS protocol is vulnerable by design and prune to security attacks by attackers [3]. These threats include:

- Cache Poisoning/DNS spoofing
- DoS and DDoS
- Masquerading
- Client flooding
- DNS Dynamic update vulnerabilities
- Information Leakage
- Government sponsored censorship

Selected threats in discussed in the following subsections

3.1.1 Cache Poisoning

When client tries to resolve the domain name and in case, DNS Server doesn't have the corresponding entries in its cache, it sends query to other DNS server to get the query resolved. In case other DNS server has maligned entries, the initial DNS server cache will be poisoned by this incorrect information. DNS spoofing is also malicious cache poisoning [2, 4]

3.1.2 DoS and DDoS Attacks

Denial of service (DoS) and Distributed denial of service (DDoS) attacks [5] are very common in DNS protocol. The weekly configured DNS servers are vulnerable to these attacks. To attack the target network, adversary sends fake DNS request to DNS server; these malicious requests are responded by DNS server with large volume of response data. To respond multiple such bogus requests, DNS server needs high computational power, which impacts genuine user and may lead to network choke.

3.1.3 Packet Interception: DNS Packet interception means
reading the DNS packets while in transit by man-in-the-middle (MITM), eavesdropping etc. As mentioned above, DNS by design uses un-encrypted and unassigned UDP protocol, which is vulnerable to such attacks. These DNS packets, while in transit on shared network, can be easily intercepted and modified by malicious attackers.

3.2 DNS security solutions using DNS secure protocols

Researchers have provided various DNS secured protocol level solutions to resolve DNS security threats. Discussion on DNSSEC and DNS Curve protocols [2, **6**] is made here.

3.3.1 DNSSEC

It is a secure DNS Protocol with new features like: data origin authentication, transaction and request authentications. DNSSEC uses digital signature and protects against MITM, DNS spoofing and cache poisoning attacks.

To get full advantage of DNSSEC protocol, both servers and resolvers must use it during DNS query/response flow, otherwise data origin authenticity and integrity will not be accomplished. Name server provides data integrity and origin of authentication whereas resolver helps in checking the signature to determine the legitimate user. DNSSEC has few limitations too. It doesn't work if there any break in continuity and the DNS records cannot be validated. It also requires several additional burdens on clients and servers (like CPU, memory and network traffic).

3.2.2 DNS Curve

DNSCurve is an alternate Public key cryptography method for the DNS protocol [8,**6**]. Elliptic Curve Cryptography (ECC) algorithm is used in DNSCurve, this algorithm has shorter key-length as well compare with DNSSEC, it has relatively high speed of encryption and decryption. DNS Curve also has few disadvantages due to which it is not publicly accepted solution.

- It fails to offer end-to-end security.

- The structure of the DNS protocol needs drastic modification
- It can be broken using Quantum computers

3.3 DNS security solutions - blockchain based solutions

There are many blockchain based solutions provided by Researchers. A few of them is discussed in this section.

3.3.1 Namecoin

Namecoin is the first fork from Bitcoin, hence it works with the same code as of Bitcoin but both operate independently as separate blockchains. The security, speed, privacy of the internet setup for DNS and identities [7] are improved using Namecoin as it uses opensource technology. It has many advantages like, hard fork of Bitcoin, features to replace both HTTPS and DNS, it solves many of the vulnerabilities in the DNS system. Another advantage is users can query the blockchain for which IP matches a domain name, instead of having to query a DNS server, which is of questionable trust. It can also query the blockchain for the hash of a public key rather than a certificate authority. Namecoin also has limitation as its consensus is based on POW, which is heavier operation.

3.3.2 Blockstack

It operates on top of the Bitcoin blockchain as a separate logical layer for naming system logic to decouple the naming system logic and consensus mechanism. It combines a DNS system with PKI. The underlying blockchain is only used for achieving consensus on the state of the naming system and the integrity of name-value data records. Block stack [5, 7] has advantage over Namecoin due its high data storage capacity, which allows the logical layer to improve and extend independently.

Block stack also does come without drawback. It has implementation limitation as it acts as a separate layer to the blockchain, which means various underlying blockchain flaws (e.g. Network delay, the time delay etc.) can impact the block stack layers.

3.3.3 Ethereum Name Service(ENS)

It helps in avoiding vulnerabilities because of its inbuilt smart contracts. Other advantage of ENS is that it doesn't have central point to attack and no intervention to mess with registration or routing. The disadvantage of ENS is system built on top of the blockchain there is no feasible method for redirecting registered names to a different address.

3.3.4 Distributed Decentralized Domain Name Service (D^{3}NS)

It provides better record storage and ownership of domain name and support backward compatibility using its discrete components [6, **8**]. To keep records in distributed structure, D^{3}NS uses Distributed Hash Table (DHT). D^{3}NS utilizes public and private key encryption for signing and verifying records. D^{3}NS also has few drawbacks. This does not solve all security issues relevant to DNS authentication and security. Exit nodes could lie or have packets inject to clients until the protocol from DNS server to client is improved. Also, this provides Domain name using mining mechanism, hence in practical, it is not a solution to solve market problem.

4. Distributed DNS Solution – DNS-Bchain

The authors propose blockchain based DNS security solution which does not alter the existing DNS protocol. The solution aims to eliminate the use of Public-Private key architecture and propose a quantum immune solution. Solution shall also have new consensus which increases the speed of validation and block addition to the chain.

It thrives for easy integration solution for the existing DNS protocol and takes advantages of immutable property of blockchain (Decentralized system). The proposed solution avoids cache ppoisoning. Proof of Concept (PoC) has been done for demonstrating the solution, which is discussed in the subsequent section

4.1 Motivation

In methods like DNSSEC and DNSCURVE which are not blockchain based DNS security protocol, the key storage as well as the structure of the DNS protocol is altered according to their specification. Also, these protocols use public-key cryptography and can be broken once quantum computing is implemented. Similarly, the blockchain based Name coin needs bit domain to have created and users have to register to utilize their service and scalability is also an issue. In Block stack it is an extra-layer on top of the bitcoin architecture, where this layer is a virtual channel. All the operations are performed on this virtual layer and this channel adds it to the blockchain. This also acts as a storage layer. The security and the time delay depend on the underlying blockchain architecture.

The proposed architecture is supposed to overcome the flaws in the existing proposals and to provide robust security solution for the DNS communications without altering original DNS protocol.

Already work done by choosing user defined port instead of default port while communicating DNS packets using blockchain [**2**]. The results show reduction in man-in-the middle attack and probability of cash poising is reduced. The proposed blockchain will be further refined using Hyperledger. The Hyperledger has been chosen over public blockchains
e.g Bitcoin or Ethereum as platform for creating private and permissioned blockchain for securing DNS. Private Blockchain will help in deploying it in Universities, Hospitals, Security organizations and various govt. hospitals without undermining their authority in using DNS blockchain solutions.

4.2 Architecture

The proposed solutions is planned to be installed on public or private cloud to take advantage of growth of number of users accessing DNS-BChain. Cloud will provide scalability on infrastructure requirements as well as high-availability and Redundancy for Disaster recovery. The Architecture diagram is shown in Figure 3.

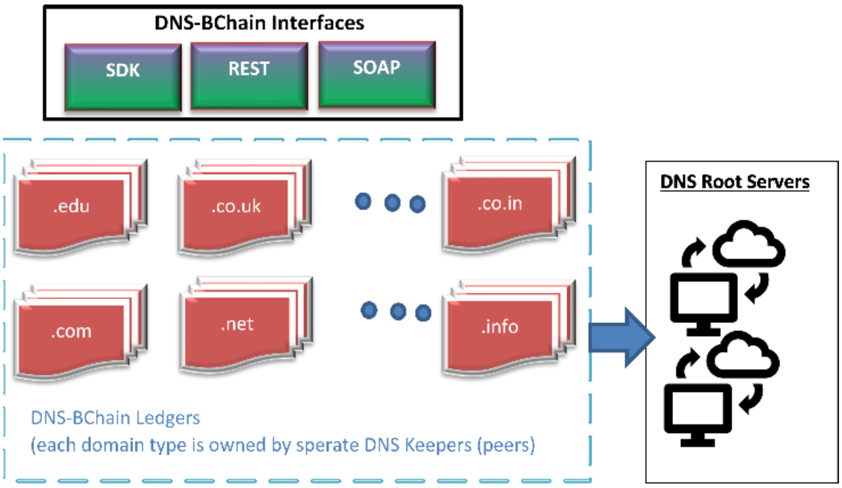

Figure 3: DNS-B Chain Architecture

For each domain type (.com, net, .edu etc.) a different ledger is proposed to store DNS Cache. Client can access DNS services of DNS-B Chain by using various proposed APIs (SDK, SOAP and REST).

4.3 DNS-Bchain - Functional Diagram

The DNS Clients call the DNS-BChain API, get IP Address, which further invokes other APIs and functions to get DNS Query resolved either from DNS or by further invoking APIs and Smart Contract to get IP Address and to update cache. If entry is available in blockchain ledger then response is given from here otherwise request is sent to other DNS server and cache is updated based on smart contract defined 17, 18].

DNS Resolution time using DNS-Chain is measured between 0.7 to 0.8 millisecond in the case where DNS entries is not available in ledger but need to be fetched via public DNS servers.

If entries are found in ledger (DNS Cache) then time taken to resolve DNS Query is around 0.5 to 0.6 millisecond. The concerned DNS-BChain functional diagram is depicted in Fig.4.

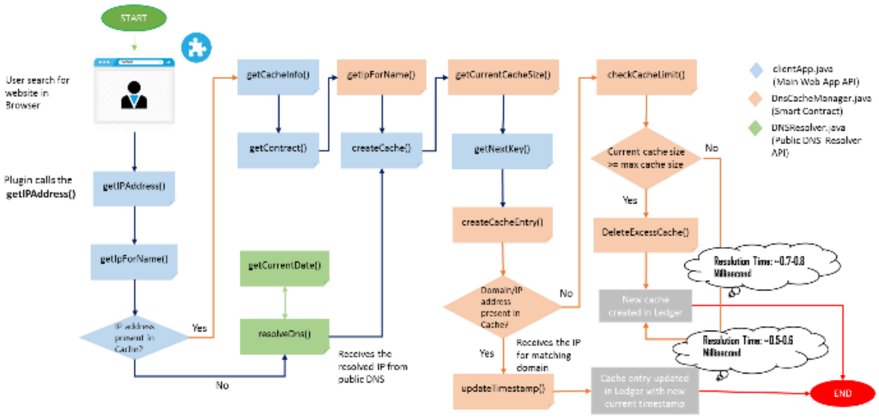
START
User search for website in Browser
Plugin calls the getIPAddress()
getIPAddress()
getIpForName()
IP address present in Cache?
Yes
No
getCacheInfo()
getContract()
getIpForName()
createCache()
getCurrentDate()
resolveDns()
Receives the resolved IP from public DNS
getCurrentCacheSize()
getNextKey()
createCacheEntry()
Domain/IP address present in Cache?
No
Yes
Receives the IP for matching domain
updateTimestamp()
checkCacheLimit()
Current cache size >= max cache size
No
Yes
DeleteExcessCache()
New cache created in Ledger
Cache entry updated in Ledger with new current timestamp
Resolution Time: ~0.7-0.8 Millisecond
Resolution Time: ~0.5-0.6 Millisecond
END
clientApp.java (Main Web App API)
DnsCacheManager.java (Smart Contract)
DNSResolver.java (Public DNS Resolver API)

Figure 4: DNS-B Chain Functional Diagram

4.4 DNS-Bchain - Cache Flush

Based on the configured value, old DNS entries are flushed out from DNS to maintain the active values only. Time can be defined as 1 month. which means, if any DNS cache entry is not used by any user within one month then it will be deleted from Cache. The concerned functional flow of DNS Cache flush function in Figure5.

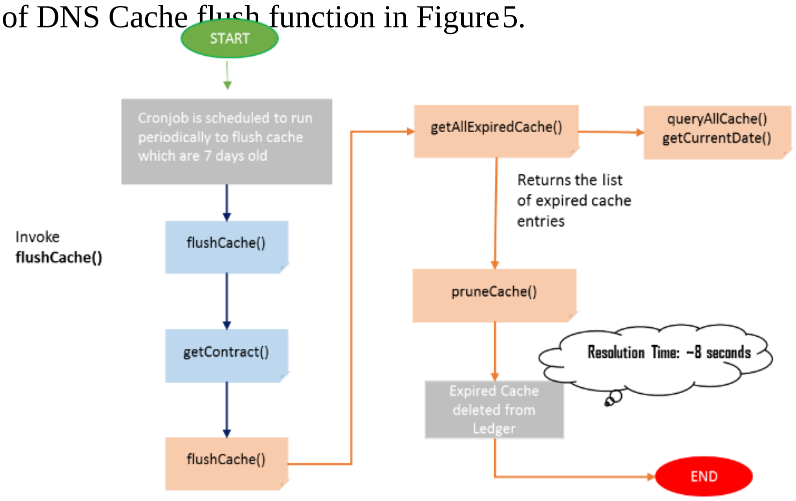
START
Cronjob is scheduled to run periodically to flush cache which are 7 days old
Invoke flushCache()
flushCache()
getContract()
flushCache()
getAllExpiredCache()
queryAllCache()
getCurrentDate()
Returns the list of expired cache entries
pruneCache()
Resolution Time: ~8 seconds
Expired Cache deleted from Ledger
END

Figure 5: DNS-B Chain Cache Flush Diagram

4.5 DNS-Bchain - Code Snippet

In this section we are detailing about code snippets. Below table is code snippet for accessing ledger for DNS entries.

Table 1: Table type styles

S. No.	Table - DnsCache.java (ledger Structure)	
	Function Name	**Description**
1	String get Domain Name()	get domain name for website
2	String get Ip Address()	get ipaddress
3	String get Ttl ()	get ttl time for each website
4	String get Record Type ()	get Record type
5	String get Record Class ()	get Record Class
6	String get Visit Date()	get Date

4.6 DNS-Bchain Consensus Mechanism - DNSCode

DNS-B Chain deploys DNS Code as consensus mechanism for DNS Query solutions. This consensus is initiated when the user queries DNS- BChain for DNS Name resolution. DNS-BChain will resolve the DNS Query and response is provided to the user. When domain queries is not resolved from DNS cache available in Blockchain ledger, multiple DNS queries (5-7 queries) requests are raised to public DNS systems like google (8.8.8.8) and others. The chain code will process these 5 responses and base on majority of similar response, ledger will be updated.

Assuming majority public DNS systems are not always poisoned at the same time but the weak point is that majority of similar response may not be true always as the public DNS systems are out of our control.

In future work, DNS servers will be hosted directly on Blockchain and these will connect root servers directly. The participating users/organization should join the DNS-BChain network for getting involved in this process. By doing this, the existing protocol is not disturbed, and the initiator and receiver can be tracked [4,5].

5. Results and Discussions

In this research the low guess ration, medium guess ratio, high guess ratio has been considered along with probability of DNS cache attack. In proposed work the DNS-BChain server receives client’s domain resolution query and process the same as per the process mentioned above, Due to Blockchain based ledger for ache, the probability of cache poisoning is reduced. Following figure is representing the comparative analysis of probability of attack considering low guess ratio, medium guess ratio, high guess ratio according to times to request URL.

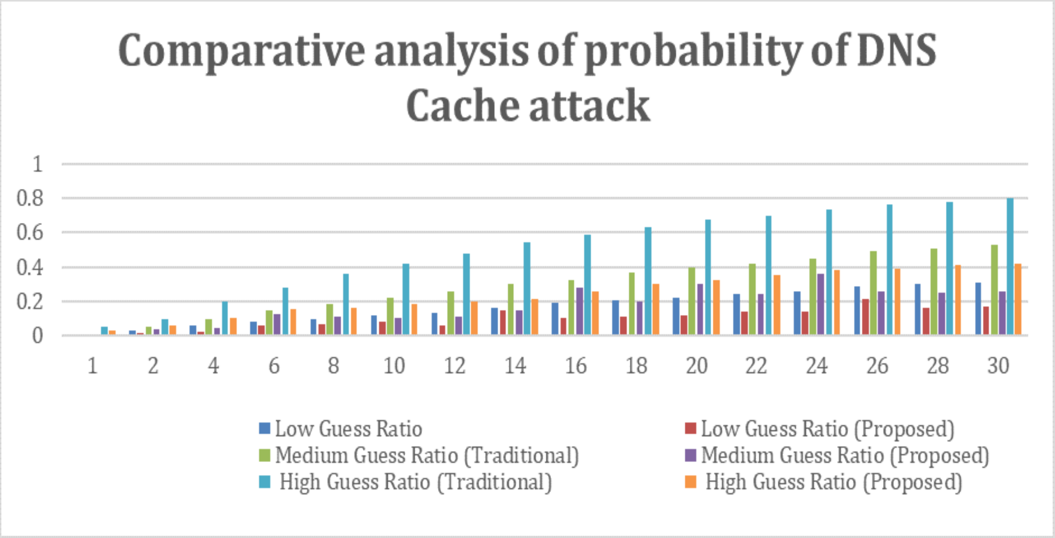

Figure 6: DNS-B Chain Result

6. Conclusions

DNS threats if exploited can lose all the information and it leads to the major security breach. Organizations are suffering in brand loss as well as huge financial loss due to DNS attacks. The breach may lead to DoS attacks, Packet interception, Name chaining and so on. Though various methods are proposed to overcome the DNS threats, there are few security issues in the proposed methodology and implementation challenges, additional software requirements at all levels or performance issues. Implementation of the above discussed methods may increase the payload affecting the speed and efficiency of the original DNS system. There is need of a new Blockchain based DNS solution to mitigate the existing flaws mentioned here.

Authors are working on Hyperledger based DNS blockchain solution named as DNS-BChain. It will deploy DNS Code (to be researched and worked-upon) as consensus mechanism for DNS Query solutions. This consensus is initiated when the local DNS Server queries DNS-BChain for DNS Name resolution. DNS-BChain will resolve the DNS Query and response is provided to the local DNS Server to communicate further to user.

From above analysis and result it is clear DNS-BChain has come up as better Solution in comparison with existing Blockchain based DNS security solutions.

Authors are working further on new consensus mechanism “Non Liner secret sharing” where the Image is shared with participants to store and for consensus verification. This may further enhance data security without affecting the DNS protocol. Authors also request future researchers to look further DNS-BChain to extend it as replacement for existing DNS solution.

Acknowledgement

Authors would like to thank Dr. Lakshmy K V, Dr. Anantha Narayanan V, Dr Chungath Srinivasan, Aswathi SV and Mrinal Verma for their guidance and help at various stages of writing this paper.

References

[1] Mukesh Kumar Bansal, M Sethumadhavan, 2020, DNS Security - Prevent DNS Cache Poisoning Attack using Blockchain, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-9 Issue-4.

[2] Mukesh Kumar Bansal, M Sethumadhavan, 2020, Survey on Domain Name System Security Problems - DNS and Blockchain Solutions” in “Com-munications in Computer and Information Science” book series (CCIS, volume 1206) , Second International Conference, FTNCT 2019, vol 1206, Chapter 50, Pages 634-647. ISBN 978-981-15-4451-4 Springer.

[3] Atkins, D., and R. Austein. ”RFC 3833: Threat Analysis of the Domain Name System (DNS), August 2004.” Status: INFORMATIONAL.

[4] Cao, J., Ma, M., Wang, X. et al. 2017, Wireless Pers Commun, 94: 1263. https://doi.org/10.1007/s11277-016-3681-2.

[5] A Brendan Benshoof, et. al. 2016, Distributed Decentralized Domain Name Service, IEEE International Parallel and Distributed Processing Symposium Workshops.

[6] Mockapetris, Paul, 2003, RFC 1034: Domain names: concepts and facilities (November 1987), Status: Standard 6 (2003).

[7] Ali, Muneeb, et al. 2016, Blockstack: A Global Naming and Storage System Secured by Blockchains, USENIX Annual Technical Conference.

[8] Wei-hong, H. U., et al. 2017, Review of blockchain-based DNS alternatives, Chinese Journal of Network and Information Security.

[9] Marios, Georgios, et al. 2012, DNSSEC vs. DNSCurve: A side-by-side comparison, DOI: 10.4018/978-1-4666-0104-8.ch012

[10] Mockapetris, Paul, 2004, RFC 1035—Domain names-implementation and specification, November 1987." URL http://www. ietf. org/rfc/rfc1035. txt (2004).

[11] P. Forte, D. Romano and G. Schmid, 2015, Beyond Bitcoin – Part I: A critical look at blockchain-based systems, PA Advice, Naples, Italy.

[12] Mohan, Ashok Kumar, and M. Sethumadhavan, 2017, Wireless Security Auditing: Attack Vectors and Mitigation Strategies." Procedia Computer Science 115, pp 674- 682.

[13] Wang, X., Li, K., et al. 2017, ConsortiumDNS: A Distributed Domain Name Service Based on Consortium Chain" IEEE 19th International Conference on High Performance Computing and Communications.

[14] Demers, A., Greene D., Hauser C., et al. 1987, Epidemic algorithms for replicated database maintenance, Proceedings of the sixth annual ACM Symposium on Principles of distributed computing. ACM, pp. 1-12.

[15] Zou, Futai, et al. 2016, Survey on domain name system security, IEEE First International Conference on Data Science in Cyberspace (DSC).

[16] Bushart J., Rossow C., 2018, DNS Unchained: Amplified Application-Layer DoS Attacks Against DNS Authoritatives, Lecture Notes in Com-puter Science, vol 11050. Springer, Cham.

[17] Jingqiang Liu, Bin Li, et al. 2018, A Data Storage Method Based on Blockchain for Decentralization DNS, IEEE Third International Conference on Data Science in Cyberspace (DSC).

[18] Kelpen K., Simo H., 2018, Privacy and Data Protection in the Domain Name System, In: Friedewald M. (eds) Privatheit und selbstbestimmtes Leben in der digitalen Welt. DuD-Fachbeitrage¨. Springer Vieweg, Wies-baden.