 ISSN: 2454-4248
09 – 12

Distributed Hash Table Implementation to Enable Security and Combat Malpractices

Dr. T. Venkat Narayana Rao
Professor, CSE,
Sreenidhi Institute of Science and Technology
Yamnampet, Hyderabad, India

K. Vivek
Student , CSE,
Sreenidhi Institute of Science and Technology
Yamnampet, Hyderabad, India

C. Pranita
Student , CSE,
Sreenidhi Institute of Science and Technology
Yamnampet, Hyderabad, India

Abstract: Security has become a global problem in any field. Everyday millions of cyber crimes are being recorded worldwide. Many of them include unethical hacking, unauthorized hampering of files and many more. To avoid such malicious practices many technologies have come up. One such security is DHT or Distributed hash tables. A recently upcoming cyber security, it helps the receiver and sender to send and receive files that are authentic. It also helps to find out if the file is manipulated or tampered by a third party or an unauthorized user by generating unique hash values. This paper mainly focuses on such distributed networks and the security they provide in order to stop malpractices.

Keywords: *Distributed Hash Value (DHT), Peer to Peer Network, Hashing Algorithms.*

*****

I. Introduction

We are becoming technologically advanced day by day. With increasing technology the need to be secured has also been increased. Online security and authentication have been manipulated and misused by many people. Cyber Security is the protection of data with the help of technologies, processes and controls which are specifically designed. An ideal cyber security aims at reducing the risk of cyber attacks and protects the system from unauthorized access and exploitation. A successful cyber security has three main components : technology, people and processes. This means that, the combined effort of all the three results in effective protection against malicious software attacks. In order to stop this we need a better system. In the last few years, an increasing number of massively distributed systems with millions of participants has emerged within very short time frames. These rapidly growing systems testify to a new era for distributed systems and their design and deployment. One such kind of technology is Distributed system i.e. Distributed Hash Table(DHT).

Distributed Hash Table (DHT)

A distributed hash table is a decentralized system which mainly works in peer to peer networks. They are upcoming cyber security systems that aim towards protecting files that are being transferred in a point to point system. DHT only has two basic operations: get data from DHT and put data into DHT. The system that DHT performs on can be either a small distributed system or large. These distributed and decentralized structures associates Hash values (keys) with some kind of input given. Information is to be evenly distributed across the network in a DHT. It is a dictionary like service over a network where it is distributed, where it provides access to a key value which is commonly shared. The key value is distributed over nodes participating in the network with great performance and scalability.

Peer To Peer Network

When there are two or more PCs connected in a network and share all the resources without depending on separate servers, termed as Peer to Peer Network. It is commonly used where there are more or less a dozen systems. Each of these acts as an independent node and has its own security system. In a P2P network one system cannot act as a client and server at the same time. To join a P2P network we only need an internet connection and P2P software. After being connected to the network we can access files on other computers which are connected in the network[3][8].

Hashing

The most common way to explain hashing is, generating a value or values of a given input string with the help of mathematical functions or algorithms. It is a way to establish security during message passing, especially when it is intended for a particular receiver. The hash algorithm or function generates a unique hash key which could be used to protect against manipulation. When a message or file is sent a unique value is generated by an algorithm. When the receiver gets the message he/she then converts the hash or decrypts the hash. He then compared the two which helps him to find out if the message is tampered or altered by a third party or an unauthorized user .We have many hashing algorithms for generating hash values like Message Digest,SHA-1,SHA 256,SHA 512,BLAKE2s,BLAKE2b.In

 ISSN: 2454-4248
09 – 12

this study Message Digest and SHA 256 algorithms are being used [1][3].

II. Architecture

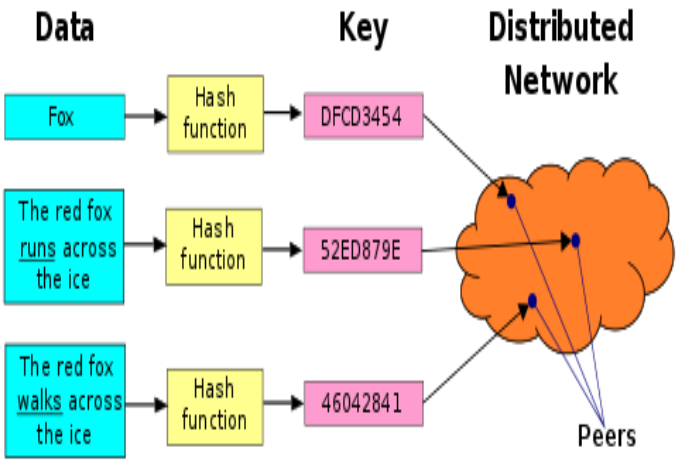

Figure 1.1 Structure of DHT

The above figure 1.1 demonstrates the structure of a DHT which can be disintegrated into a few primary parts. The establishment is a dynamic keyspace, for example, the arrangement of 160-piece strings. A keyspace parceling plan parts responsibility for keyspace among the taking an interest hubs. An overlay organize then associates the hubs, enabling them to locate the proprietor of some random key in the keyspace.

When these segments are set up, an ordinary utilization of the DHT for capacity and recovery may continue as pursues. Assume the keyspace is the arrangement of 160-piece strings. To list a document with given filename and information in the DHT, the SHA-1 hash of filename is created, delivering a 160-piece key k, and a message put(k, information) is sent to any hub taking an interest in the DHT. The message is sent from hub to hub through the overlay organize until it achieves the single hub in charge of key k as determined by the keyspace division. The hub at that point stores the key and the information. Some other customer would then be able to recover the substance of the document by again hashing filename to create k and requesting that any DHT hub discover the information related with k with a message get(k). The message will again be directed through the overlay to the hub in charge of k, which will answer with the put away information. [1][2]. The keyspace parceling and overlay arrange segments are depicted underneath with the objective of catching the chief thoughts regular to most DHTs; numerous plans vary in the subtleties.

The primary objective behind any digital assault is to increase unapproved access to something of significant worth, this could be: information, protected innovation, hidden system, or the clients' PCs. All the current digital security issues are not new, but rather they have taken progressively risky structures either on account of the accessibility of new advancements , the approaches are :

1-APT: Progressed Steady Risk, as a rule state-supported programmers, who can access a system undetected and remain as such for at some point. Ordinarily, the objective of the customary digital assaults is get to an objective and leave, this objective could be information, data, licensed innovation, and so on. The other objective is to get in and remain in, so the aggressors must get in undetected and change their strategies always to dodge identification by ordinary Interruption Recognition devices. Stuxnet is a genuine case of APT[6][8].

2-Ransomware: This is an old issue that got another breath of life. The first ransomware was distinguished in 1989, so it's anything but another issue. The ongoing ascent of ransomware is primarily because of computerized coins, which can't be followed, for example, BitCoin ending up increasingly standard. Ransomware more or less, is a bit of malware, which the assailant traps the unfortunate casualty into executing, when it is executed it begins to specifically encode essential documents, for example, *.doc, *.xls, *.pdf, and so on. When the unfortunate casualty pays the payment sum, the aggressor sends the injured individual the way to decode the files[6].

3-Application Security: These are the product coding imperfections that assailants use to hack applications, and could prompt full framework takeover. For instance, SQL Infusions; which is an aggressor where the assailant traps the application to execute directions they control. A large portion of the untouched information ruptures could be connected to a SQL Infusion; for example TJX 94M Visas - 2006 , Heartland Installment Frameworks - 130M Charge cards - 2012, and so forth. There are different defects, for example, cross-site scripting, different sorts of infusion blemishes, validation and approval issues etc.

III. DHT as a Proposed System

A DHT can be seen as a lexicon benefit disseminated over a system as it gives access to a typical shared key esteem appropriated over taking an interest hubs with extraordinary execution and versatility. To achieve this goal, the concept of consistent hashing is used. A key is passed through a hash algorithm that serves as a randomization function. This ensures that each node in the network has an equal chance of being chosen to store the key/value pair. These systems have been created in order to verify whether the sent file is manipulated or not.

It is difficult to lose database as copies are replicated across the network. Whenever there are any big changes across the network, it is resilient to them. According to specific configuration options, all the data is distributed

 ISSN: 2454-4248
09 – 12

automatically. A single node removed doesn't affect the entire network.

A. Preprocessing

In this phase the image is divided into number of parts as per the requirement. It is necessary to specify the path of the image and after specifying the path we need to input the number of rows , columns , length and the breadth then the image loaded is partitioned into number of parts as per our response . Later these are automatically stored in the default directory which is chosen .

Step 1. Create class Image Split
Step 2. Give the file path that is to be read.
Step 3. Read the image.
Step 4. Read rows and columns into some variables.
Step 5. Read height and width into some variables.
Step 6. Image array is initialized and image chunks are loaded into the array.
Step 7. Image Splitting is Completed.
Step 8. By using the iterative loop the split parts are converted into the image format.
Step 9. They are automatically stored in default directory set which is predetermined.

B. Generating Hash

The outcome i.e. image files which are generated from the splitting acts as an input for the current method . Here each part is considered separately and employ hash function named SHA 256(Secure Hash Algorithm) for generating hash values which generates an unique value for each file correspondingly. It generates a 32 byte code .When a file is sent to the peers we need to send the files along with their respective hash values[4][5]. The following is the process involved.

Step 1. Create class SHA.
Step 2. Specify which algorithm to generate.
Step 3. Specify the path of the file.
Step 4. Initialize Bread.
Step 5. Read a file only If Its size is non zero.
Step 6 .Call- The Digest Method.
Step 7. Generate Hash value for each file.
Step 8. Send Hash values to Peer with the file.

```
C:\Users\kapa\Desktop\project>java Sha
SHA : b1b2b4588f6e4c899b5652997f6e0986b767eb72fb997511363adb4712eef44c
```

Figure 2.1 Sample Code

The Above Figure 2.1 shows a sample 32 Byte Code generated by Hash Function SHA 256 for a file.

C. Verification

When the peer receives the file he/she must use the Hash Function in order to ascertain whether any third party has accessed the File or not. In the event of change there would be change in the Hash Values due to change in the resolution of the image. So if there is any change in the hash value generated to that of received, it reflects that the particular image file is corrupted and it is very difficult to regenerate the whole image file as each part has got different 32 byte Hash Code. In this phase Message Digest Algorithm (MD5) is employed as shown in the algorithm below [6][7].

Step 1. Create Class Verify.
Step 2. Generate the Hash value.
Step 3. Assign File Path.
Step 4. Call Message Digest Method .
Step 5. Generate Hash Value for a File.
Step 6. Compare Two Values.
Step 7. If Equal then Safe , If not then it is Corrupted file.

IV. Applications

Numerous ideas for applications of DHTs have come up in the past few years. Though many of them are still at developing stage, they prove to be promising technology in the future that will solve many of the security problems . Few applications are :

1. BTDigg, which is a BitTorrent based search engine.
2. Tox, which is an instant messaging which will intend to work as Skype.
3. JXTA, which is an open source platform.
4. FAROO which is a Search Engine.
5. Secured File Sharing.

V. Conclusion

DHT is not just a single field, but contains many algorithms such as MD5, SHA1, SHA256 etc. Providing both fault tolerance and scalability, DHTs aim to provide almost accurate results. Although it is not put to use properly yet, DHTs are expected to take over cyber security in near future. With technologies like DHTs and block chain, cyber security would become the most secured entity. Hence, it is safe to conclude that lot of improvements in future regarding cyber security and DHT are unfolding for a safe digital domain.

References

[1]. Increasing DHT Data Security by Scattering DataAuthors: Bryan Mills ; Taieb Znati
[2]. Fabius Klemm ; Sarunas Girdzijauskas ; Jean-Yves Le Boudec ; Karl Aberer "On Routing inDistributed Hash Tables".
[3]. R.Schollmeier "A definition of peer-to-peernetworking for the classification of peer-to-peerarchitectures and applications ".
[4]. Zhenqi Wang ; Lisha Cao "Implementation andComparison of Two Hash Algorithms ".
[5]. WanzhongSun; HongpengGuo ; HuileiHe ; ZibinDai " Design and optimized implementation of theSHA-2(256, 384, 512) hash algorithms."

ISSN: 2454-4248
09 – 12

[6]. Komal Gandhi "Network security problems andsecurity attacks"
[7]. Zhao Yong-Xia ; Zhen Ge "MD5 Research".
[8]. G. Kwon ; K.D. Ryu "An efficient peer-to-peerfile sharing exploiting hierarchy and asymmetry".