ISSN 2047-3338

A Survey and Overview on Distributed Hash Table, its Implementation, Applications and Security Techniques

Noor Zaman, Kamal Ahmad, Bilal Zafar, ShafiUllah and Zaheer Aslam

CECOS University, Hayatabad, Peshawar, Pakistan
Ministry of Information Technology, Peshawar, Pakistan

***Abstract*– This paper presents a survey and overview on Distributed Hash Table, its implementation, Applications and Security techniques. This paper presented on the literature for dealing with Security concerns associated to malevolent Nodes in structured Peer-to-Peer Networks, furthermore identified as Distributed Hash Tables. We explain achievable resistance and protection against several distinguished Attacks and analysis their advantages and disadvantages in the form of tables. We also verify how complicated it is to protect such type of a System in an unfavorable situations, as we were incapable to find a System capable to survive all the present and studied Attacks.**

***Index Terms*– Overview, Applications, Attacks, Security and Implementation**

I. INTRODUCTION

A Distributed Hash Table (DHT) is a group of a decentralized Distributed System, which supplies a Lookup Service analogous to a Hash Table; and whichever contributing Node can well recover the value connected with a prearranged key. Dependability for preserving the Mapping from keys to value is circulated between the Nodes, in such a method that an alteration in the set of contributor causes a minimum quantity of disturbance. This allocates a Distributed Hash Table to size to exceptionally great numbers of Nodes and to control frequent Node appearances, exit, and malfunction.

Distributed Hash Tables form an Infrastructure that can be exercised to make further composite Services, such as Anycast, cooperative web caching, Distributed file System, domain name Services, instant messaging, multicast, as well as Peer-to-Peer file Distributed and content Distributed Systems. Prominent Distributed Networks that utilize Distributed Hash Tables include Bit Torrent Distributed tracker, the Coral content Distribution Network, the Kad Network, the storm BotNet etc [1]. The concept of a decentralized Lookup Service is very helpful for many Distributed applications [2] – [6]. Such a service offers the essential procedure lookup (k), which precedes data connected to the key (k) nearly all-General literature to put into operation this roles is the use of structured Peer-to-Peer systems, and identified as Distributed Hash Tables (DHT) [7] – [10].

The very demanding feature of Distributed Hash Tables is that it is extremely complicated to compose them tolerant to the existence of malevolent and perhaps conspiring Nodes in an open situation such as the Internet, where jointly inelegance parties are allowable to connect the system.

In this paper, we survey the study in the region of Distributed Hash Table Security. We employ the conventional method of arguing potential Attacks and the related resistance as that is the method taken by very of the related literature. Although we have made an attempt to cover up the much related apprehensions, this method constantly requires the disadvantage that there is no way to assurance that all possible Security threats are measured. We categorize Attacks as either General or specific. General Attacks are those that affect not only to Distributed Hash Tables but also to many kinds of Overlay Networks. General Attacks are frequently used to make possible the implementation of Attacks particular to the architecture of the Attacked Overlay. Our most important involvement is that we argue many of the methods planned to pact with adversarial Nodes in Distributed Hash Tables and evaluate their advantages and disadvantages.

II. GENERAL ATTACKS

In this part we explain Attacks that are not particular to Distributed Hash Tables, however Overlay Networks in General. It is very significant to learn them as Distributed Hash Tables are defenseless to them and they can be used to assist the implementation of further particular Attacks.

Sybil Attack

This Attack was first study by Donceur [11]. It develop the information that in a Distributed System, distant entities are apparent as informational abstraction that can be describe as identities. If the System stops working to promise that every identity Refers to a single physical entity, a malevolent Node might generate a huge amount of identities and control the Overlay Network by tricking the protocols and destabilize mechanisms found on redundancy. The very essential ending of Donceur's study is that in a P2P System, containing a reasonably central, reliance authority to question identities is the single realistic way to promise a one-to-one communication among the identities and the physical entities that function the contributing Nodes. Table 1 summarize and offers a contrast of the resistance in opposition to Sybil Attacks.

Attrition Attacks

Maniatis et al. [12] explain an attrition adversary as one who tries as to avoid clients of the System from achieving timely Service. They study attrition Attacks and suggest a number of methods for protecting against them. They relate their outcomes to a digital protection System, but the outcomes are related for all P2P Systems, together with Distributed Hash Tables. To our understanding, there are no consequences on attrition Attacks particular to Distributed Hash Tables.

Here we explain three types of attrition Attacks:

- *Pipe Stoppage:* This is the conventional Denial of Service Attack in which the sufferers Network links are saturated due to a very high demand rate.
- *Anomalously elevated rates of requirements*: In this case, the Attacker sends specifically formed requests at a rate that does not saturate the sufferers Network links, but does saturate other resources. The rate at which this requests are sent are higher than a normal requester's, so they may finally be distinguish and filtered.
- *Apparently harmless rates of requests:* The Attacker, with a systematic accepting of the protocols, sends requests at a regular rate, but the requests are watchfully crafted so that they saturate victim's resources.

This attrition Attack is the very complicated to notice. To preserve from these Attacks, the subsequent methods are planned:

Effort balancing: The scheme is to compose the attempt desirable by a requester to offer a request similar to the effort essential to Service it. This avoids Attackers from transferring 'cheap' requests and after that failing or pays no attention to the response.

Rate limitation: Peers must practice requests no quicker than necessary as a substitute of as quick as achievable. This can efficiently slow down Attacks to the point of make them not practical.

Admission control: In order to submit an application to the rate restriction method, it is essential to refuse or drop a few requests. Here we believe random drops, session-base classification [13] and reputation-based classification.[14].

Redundancy: This avoids rejecting admission to a specific Service by attacking only one node.

Compliance Enforcement: A requester ought to offer evidence that it has exercise a response. The design is that dealing out a response more often entail a cost that an Attacker would usually want to circumvent.

Resynchronization: Synchronization must be circumventing in process that need numerous peers, such as waiting for a busy server or, in the case of a Distributed Hash Table, circulation of update to routing tables. This kind of synchronized operations establishes insecurity and amplifies the load on the contributing nodes.

III. SPECIFIC ATTACKS

A. Eclipse Attacks

Nodes in Overlay Network having links to other peers called neighbor. If an Attacker organizes a great part of the neighbor of accurate Nodes, then the proper Nodes can be "eclipsed" by the malevolent Nodes by avoiding messages from getting the accurate nodes. This attack is also recognized in the literature as routing table poisoning. It is tricky to organize this attack as general or specific, because the entire overlay networks are susceptible to it and capable of to be exercise to carry out more specific attacks. On the other hand, we have determined to categorize it as a specific Attack as the ways to perform such an attack in a distributed hash table are very particular to the formation of the Distributed Hash Table.

Sit and Morris [15] were the first to study this attack in the perspective of Distributed Hash Tables, and they affirm that systems in which the neighbor does not have particular necessities are the very exposed to this type of attack all the way through erroneous routing updates. In exacting, the argue that systems that exercise network proximity information as the major or sole criterion to choose neighbor, are prone to eclipse attacks as a malevolent node is more likely to be acknowledged as a neighbor by a accurate node.

B. Routing and Storage Attacks

There are a lot of means by which a malevolent node know how to avoid a lookup demand from being successful e.g. an attacker might reject to onward the lookup request, or it might forward it to an inaccurate, malevolent node, or it might as well make believe to be the node accountable for the key. An additional option for an attacker is to route appropriately, although reject the survival of a suitable key or provided that illogical data as a reply. In this part we study a number of resolutions that have been planned to this kind of problems.

This system has the subsequent characteristics [17]:

With high possibility, at whichever time, an arbitrarily huge fraction of the nodes know how to locate an arbitrarily huge fraction of the keys.

Lookups get O(log n) time and need O (log3n) messages.

Every node sustains pointers to O (log3n) further nodes.

Each node preserve O(log n)keys.

Key inclusion get O(log n) time.

IV. DHT IMPLEMENTATION

Most remarkable differentiation meets in realistic cases of Distributed Hash Table implementation containing at least the subsequent points:

i) The address space is a limitation of Distributed Hash Table. A number of true world Distributed Hash Table use 128 bit or 160- bit key space.

ii) Some real-world Distributed Hash Table use Hash roles other than SHA-1.

iii) In the real world the key Κ might be a Hash of a file's content relatively than a Hash of a file's name to offer content-addressable storage, so that the renaming of the file does not avoid users from discovering it.

iv) Some Distributed Hash Table may as well distribute substances of dissimilar kinds, e.g., key Κ might be the node ID and connected data may perhaps explain how to make contact with this node. This permit Publication-of-Presence information and frequently used in IM functions, etc. In the simplest case, ID is now an accidental number that is straightforwardly used as key (Κ), in several Distributed Hash Table, distributing of nodes ID s is as well exercise to optimize Distributed Hash Table processes.

v) Redundancy is able to be further to progress consistency.

The *(k, data)* key pair be able to be stored in further than one node related to the key. Typically, relatively than choosing just one node, real world Distributed Hash Table algorithms choose appropriate nodes, with being implementation; specific factors of the Distributed Hash Table. In a number of Distributed Hash Table plan, nodes be in agreement to handle a definite key space collection, the amount of which might be selected dynamically, relatively hard-coded.

vi) Several sophisticated Distributed Hash Table like kademalia carry out iterative lookups all the way through the Distributed Hash Table first in order to choose a set of appropriate nodes and drive messages *put (Κ, data)* only to those nodes, therefore radically falling ineffective traffic, as published messages are only send to nodes that look appropriate for storing the key Κ; and iterative lookups cover up just a little set of nodes relatively than the whole Distributed Hash Table, dropping ineffective forwarding. In such Distributed Hash Table, forwarding of *put (Κ, data)* messages might only happen as part of a self-healing algorithm [1].

Applications [18]: Anything that need Database, FSes, storage archrivals.

i). Web serving, caching
ii). content distribution
iii). Query and Indexing
iv). Naming system
v). Communication primitives
vi). Chat services
vii). Application layer multicasting
viii). Event notification services

V. CONCLUSIONS

Here we talk about some renowned security threats face by Distributed Hash Table and have evaluated numerous methods planned to resolve or moderate them. The range of the planned results, in addition to their incapability to resolve efficiently each and every calculated problem, illustrate how complex it is to safe a DHT system in an aggressive surroundings with malevolent Nodes. In the future, we sketch to deal with the trouble of decentralized lookup services using a method depends on game theory in which the adversial model depends on selfish conduct. We would like to guess If it is probable to produce a decentralized lookup services in which each and every one nodes perform properly as it is in their own selfish conduct to do so. Furthermore we want to estimate if this kind OS system is capable to survive attacks carry out by malevolent node and not essentially selfish nodes such as the ones reviewed in this paper.

REFERENCES

[1]. Free Wikipedia: http://www.wikipedia.com

[2]. Min Cai, Ann Chervenak, and Martin Frank. “A peer-to-peer replica location service based on a distributed hash table”, In SC ’04: Proceedings of the 2004 ACM/IEEE conference on Supercomputing, Washington, DC, USA, 2004. IEEE Computer Society.

[3]. Frank Dabek, Frans M. Kaashoek, David Karger, Robert Morris, and Ion Stoica. Wide-area cooperative stor-age with cfs. In SOSP ’01: Proceedings of the eighteenth ACM symposium on Operating systems principles, volume 35, pages 202–215, New York, NY, USA, Dec 2001. ACM Press.

[4]. Peter Druschel and Antony Rowstron. Past: A large-scale, persistent peer-to-peer storage utility. In HOTOS ’01: Proceedings of the Eighth Workshop on Hot Topics in Operating Systems, Washington, DC, USA, 2001. IEEE Computer Society.

[5]. Venugopalan Ramasubramanian and Emin G. Sirer. The design and implementation of a next generation name service for the internet. In SIGCOMM ’04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications, volume 34, pages 331–342, New York, NY, USA, October 2004. ACM Press.

[6]. Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Schenker. A scalable content addressable network. In Proc. SIGCOMM Conf., pages 161–172, 2001.

[7]. Antony I. T. Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In Proc. Middleware Conf., pages 329–350, 2001.

[8]. Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, and Hari Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. IEEE/ACM Trans. Netw., 11(1):17–32, 2003.

[9]. Ben Y. Zhao, Ling Huang, Jeremy Stribling, Sean C. Rhea, Anthony D. Joseph, and John D. Kubiatowicz. Tapestry: A global-scale overlay for rapid service deployment. IEEE Journal on Selected Areas in Communications, 2003.

[10]. John R. Douceur, “The sybil attack. In IPTPS ’01: Revised Papers from the First International Workshop on Peer-to-Peer Systems, pages 251–260, London, UK, 2002. Springer-Verlag.

[11]. Petros Maniatis, TJ Giuli, Mema Roussopoulos, David S. H. Rosenthal, and Mary Baker. Impeding attrition attacks in p2p systems. In EW11: Proceedings of the 11th workshop on ACM SIGOPS European workshop: beyond the PC, page 12, New York, NY, USA, 2004. ACM Press.

[12]. Ludmila Cherkasova and Peter Phaal. Session-based admission control: A mechanism for peak load management of commercial web sites. IEEE Transactions on Computers, 51(6):669–685, 2002.

[13]. Michal Feldman, Kevin Lai, Ion Stoica, and John Chuang. Robust incentive techniques for peer-to-peer networks. In EC ’04: Proceedings of the 5th ACM conference on Electronic commerce, pages 102–111, New York, NY, USA, 2004. ACM Press.

[14]. Emil Sit and Robert Morris. Security considerations for peer-to-peer distributed hash tables. In IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems, pages 261–269, London, UK, 2002. Springer-Verlag.

[15]. Miguel Castro, Peter Druschel, Ayalvadi Ganesh, Antony Rowstron, and Dan S. Wallach. Secure routing for structured peer-to-peer overlay networks. SIGOPS Oper. Syst. Rev., 36(SI):299–314, 2002.

[16]. K. Hildrum and J. Kubiatowicz. Asymptotically efficient approaches to fault-tolerance in peer-to-peer networks, 2003.

[17]. Atul Singh, Tsuen-Wan "Johnny" Ngan, Peter Druschel, and Dan S. Wallach. Eclipse attacks on overlay networks: Threats and defenses. In IEEE INFOCOM 2006, Barcelona, Spain, April 2006.

[18]. T. Condie, V. Kacholia, S. Sankararaman, J. Hellerstein, and P. Maniatis. Induced churn as shelter from routingtable poisoning, 2006.

Table 1: Summarizes and offer a comparison of the defenses against Sybil Attacks

Authors	Castro et al [4]	Danezis et al .[8]	Dinger and Hartenstein [9]	Wang et al. [18]
Technique	Certificates signed by a trusted authority, possibly paid.	Use of bootstrap graph.	Distributed registration counts.	Use of physical Network characteristics to recognize nodes.
Advantages	To allow excellent control on those who are allowable to connect to the system and help to make the security of other protocols better.	Does not Putting the barriers to join the system. Decentralized.	Does not Putting the barriers to join the system. Decentralized	Does not Putting the barriers to join the system.
Disadvantages	To introduce administrative and processing overhead. Putting barriers to legal nodes and try to join the system.	To introduce Significant overhead and must not shown to a limit beyond 100 nodes.	Does not Putting real assurance of Sybil security. To introduce Introduces clear possibilities for new attacks.	Depends on Network Measurement which can vary with time for the similar node and therefore failing to offer a regular identity. Changes to the Network Measurement infrastructure may invalidate to recognize all nodes.

.

Table 2: Summarizes and offers a comparison of the defenses against Eclipse Attacks

Authors	Castro et al. [4]	Hildrum and kubiatowiccz [14]	Singh et al. [18]	Condie et al. [6]
Technique	Use of two Routing Tables. One Optimized with Network Measurement and one constrain utilized in case of a test Malfunction.	Use of redundant Routing table entries depends on Network Proximity.	Management of the in degree and out degree of Overlay Nodes through Unidentified auditing.	Reorganizing of Optimized table entrances and induced churn.
Advantages	Permit the exercise of Proximity Routing in usual cases. Do not suppose the Existence of Services that might be complex to execute.	Simplicity. Permit the use of Proximity routing.	Permit the use of Proximity routing. Do not suppose the continuation of Services that might be hard to execute.	Deal with the difficulty of Progressive Poisoning of the Routing Tables. Permit the use of Proximity routing.
Disadvantages	Do not address the concern of Progressive Poisoning of the Routing Tables. The Routing malfunction test Do not look to be very correct and is extremely susceptible to Nontrivial factors.	Depends on well-organized and precise Network Distance measurement, that might be complex to implement existing in practice.	Has not been exposed to level to systems with further 1020. Nodes with the planned anonymzing methods. Experimental outcomes illustrate that the system is efficient simply when the degree bound is small. This outcomes in an enhance of the lookup time in the nonexistence of attacks.	Induced churn might not be adequate in various systems. establishing a significant overhead. The Routing malfunction test Do not look to be very correct and is extremely susceptible to Nontrivial factors. Though this is lessen with the reset of Routing Tables. The management of the dependence unpredictability service might be difficult in practice.

Table 3: Comparison of Defenses against Routing and Storage Attacks

Authors	Castro et al. [4]	Hildrum and kubiatowiczz	Saia et al
Technique	Use of two Routing Tables. One Optimized with Network Measurement and one constrain utilized in case of a test Malfunction.. The constrained Tables is used with redundant Routing over disjoint paths	Use of redundant Routing table entries depends on Network Proximity. Routing uses wide paths, taking advantage of the redundant table entries.	Butterfly Network of super Nodes. genuine Nodes are mapped to numerous super Nodes This outcome in Routing using Wide paths.
Advantages	Permit the exercise of Proximity Routing in usual cases. Do not suppose the Existence of Services that might be complex to execute.	Wide path Routing look Very consistent than disjoint paths Routing. Benefits from Proximity Routing.	Wide path Routing look Very consistent than disjoint paths Routing. Benefits from Proximity Routing.
Disadvantages	The Routing malfunction test Do not look to be very correct and is extremely susceptible to Nontrivial factors. The amount of necessary disjoint paths may be huge as a single malevolent node entirely overthrow the path in which it is integrated.	Depends on well-organized and precise Network Distance measurement, that might be complex to implement existing in practice.	As the super node generalization are cliques of definite Nodes communication among super Nodes might be reserve concentrated. Does not utilize proximity Routing. Addresses arbitrary node elimination attacks, however does not address intimidation such as the eclipse attack. No investigational outcomes are offered.