International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 16 (2014), pp. 1719-1726

http://www. irphouse.com

Security and Fault Tolerance Enhancement in Hybrid P2P

Vivek Kumar Prasad

Computer Science and Engineering Department,
Nirma University, Ahmedabad.

Abstract

In the Internet today, computing and communications environments are significantly more complex and chaotic than the classical distributed systems, lacking any centralized organization or hierarchical control. Which has given rise to the emerging field of Peer - to - Peer (P2P) network as well as the Hybrid P2P. In this paper I have focused much on the hybrid P2P network, were a central server exists to perform certain administrative functions to facilitate P2P services. Security is an inevitable issue to be considered in the Hybrid P2P systems. In this paper I have proposed security using Advanced Encryption Standard (AES) in the Hybrid peer to peer by using different keys i.e 128, 192 and 256 bit keys based upon the hop distances between the two nodes in the network, so that the time required for doing encryption or decryption can be reduced and as a result the overall communication delay can be minimized, I have also proposed an idea for the implementation of the fault - tolerance, based upon the hop distances in the network, which will be used when a peer goes down or is disconnected from the network, then another peer will be available to the nearby nodes which continue to give its own services on behalf of the peer which has been failed. The overall purpose of this paper is to analysis the shortest distances between the nodes and based upon the distance the security key will be selected and the extra node will be placed to nearby nodes, so that if the particular node fails, another will be present to give the services (Fault - Tolerance).Thus the communication cost can be minimised to an extent.

Keywords: Peer to Peer, Hybrid Peer to Peer, Advanced Encryption Standard, Fault Tolerance, Hop Distance

I. Introduction

Distributed system has been developed as a platform for huge computations. Reliability is one of the prominent issues in such systems. Many studies have been recently done to improve the security in P2P and fault tolerance. In a hybrid P2P network, a central server exists to perform certain administrative functions to facilitate P2P services. For example, in Napster, a server helps peers to search for particular files and initiate a direct transfer between the clients [1]. Only a catalogue of available files is kept on the server, while the actual files are scattered across the peers on the network. Compared to the hybrid P2P architecture, the pure P2P architecture is simpler and has a higher level of fault tolerance. On the other hand, the hybrid P2P architecture consumes less network resources and is more scalable than the pure P2P approach. P2P file sharing networks are an important and fast growing part of the Internet communication. Millions of people are using P2P networks today to share text, software, audio, and video files stored on their computers [2]. Even the use of P2P file - sharing software can raise serious security issues [3, 4] as the sensitive personal information can be at risk due to improper usage by users. At the same time, file - sharing technology is largely user controlled, which is sometimes beneficial but hard to regulate. In addition, to fully realize the potential of the P2P paradigm, such systems must be able to support an open environment where mutually distrusting parties with conflicting interests are allowed to join and get the response. In such environments, where there are many diverse parties without a pre - existing trust relationship, the security is particularly important and nontrivial. High availability [5] is a desired feature of a dependable distributed system and replication is a well - known technique to achieve fault tolerance in distributed systems, thereby enhancing availability. The rest of the paper is organized as follow.In section II the various definitions which are used herein are briefly discussed and in section III gives the details about the problem statements and in section IV illustrates the proposed solution which is then followed by the conclusion.

II. Definitions

Security Threats in Hybrid P2P

The P2P applications need the firewall to open a number of ports in order to function properly. Each open port in the firewall is a potential avenue that attackers might use to exploit the network. It is not a good idea to open a large number of ports in order to allow for P2P networks. Propagation of

malicious code such as viruses: As P2P networks facilitate file transfer and sharing, malicious code can exploit this channel to propagate to other peers. When a file is downloaded using the P2P software, it is not possible to know who created the file or whether it is trustworthy. In addition to the risks of viruses or malicious code associated with the file, the person downloading the file might also be exposed to criminal and/or civil litigation if any illegal content is downloaded to a company machine. Also, when downloading via a P2P network, it is not possible to know what peers are connected at any one time and whether these peers are trustworthy or not. Untrusted sources induce another security threat. Then, Vulnerability in P2P software: Like any software, P2P software is vulnerable to bugs. As each peer is both a client and a server, it constantly receives requests from other peers, and if the server component of the P2P software is buggy, it could introduce certain vulnerabilities to a user's machine. Intruders could exploit this to spread viruses, hack into a machine, or even launch a denial of service attack.

Advanced Encryption Standard

The Advanced Encryption Standard (AES) is a key - iterated block cipher [6] that has been approved by the U.S. Government. The AES is originally the Rijndael Algorithm, created by Joan Daemen and Vincent Rijmen, as a part of the US National Institute of Standards and Technology (NIST) requirement to have an Advanced Encryption Standard to replace the obsolete Data Encryptions Standards (DES) algorithms. It is fast w.r.t the software as well as in hardware too. The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input, called the plaintext, into the final output, called the cipher text. The number of cycles of repetition are 10 cycles of repetition for 128 - bit keys, 12 cycles of repetition for 192 - bit keys and 14 cycles of repetition for 256 - bit keys. The key length used in the encryption determines the practical feasibility of performing a brute - force attack, with longer keys exponentially more difficult to crack than shorter ones. Brute - force attack involves systematically checking all possible key combinations until the correct key is found and is one way to attack when it is not possible to take advantage of other weaknesses in an encryption system.

Fault tolerance in Distributed system (P2P)

The system has to be tolerant against network and site failures. This is achieved through replicating object - states to other sites. All objects need to be available for access at any available site at all times even in presence of arbitrary system degradation scenarios.

Distances between the nodes

Accurate prediction of network distance between information collection node and websites distributed in different locations is the basis for collaborative information collection, and also localize the process of collecting sites, thereby reduce the network distance overhead, improve collection efficiency, reduce network load, enhance fault tolerance capability [7], which is of great significance.

1) Round - Trip Time (RTT): The time delay from the beginning of sender sending data to receiving confirmation from the receiving end (the receiver sends an acknowledgment immediately after receiving data) is a total RTT.
2) Hop number (Hop Count): the total number of devices passed through by a specific data (packet).

III. Problem Statements

1. Longer key lengths are better, but only up to a point. AES will have 128 - bit, 192 - bit, and 256 - bit key lengths. This is far longer than needed for the foreseeable future. In fact, we cannot even imagine a world where 256 - bit brute force searches are possible. It requires some fundamental breakthroughs in physics and our understanding of the universe and even more is the key length the more time it will take to decrypt it. In this paper I am assuming all the peers are enough hardware and software support to deal with the AES. It has been observed that increase in the key length leads to more processing time, thus hacker have more number of options to destroy the model [8]. So we need to use the various keys i.e 128 - bit, 192 - bit, and 256 - bit key lengths in a wise manner, I mean to say that where ever it suits to the requirement.
2. The system has to be tolerant against network and site failures. This is achieved through replicating object - states to other sites. Replication for fault tolerance basically comprises of four components [9]: -

i. Information Policy

The information policy is responsible for collecting the system state information. In local state information, state information of neighbouring nodes is collected while in global statistics, state information of all the nodes in the system is collected for better scheduling decisions

ii. Transfer Policy

The transfer policy decides the state of the node that is lightly loaded or heavily loaded. Widely used transfer policy is threshold policy.

iii. Selection Policy

The selection policy selects a job to be transferred. There are several factors to consider in the selection of a job: (a) the overhead incurred by the transfer should be minimal; (b) the number of location - dependent system calls made by the selected job should be minimal; (c) the selected job should be long lived so that it is worthwhile to incur the transfer overhead (d) a job should be selected only if its response time will be improved upon transfer.

iv. Location Policy

The location policy is responsible for selecting the best node among all the available nodes. It finds suitable destination node to share the load. The factors to be considered while selecting the node may include resource availability, the availability of the service(s) required for the proper execution of the migrated process, etc. The selected node should have the correct environment to run the process. Various location policies have been presented in the literature as in [10] [11] that are random policy, polling (Probing) policy : threshold, greedy, and shortest, negotiation policy: bidding and drafting, low load policy, broadcast a query, main and secondary locations policy. Explaining all the algorithm is beyond the scope of this paper.

Replication is a widely used technique to guarantee the availability and dependability of large scale distributed systems in the presence of faults is replication and implies the use of more services or components performing the same function. Whenever a replicated entity encounters a failure another replica is switched on and takes its place. Data and service replication is a reliability improvement technique used in many types of distributed systems. By replicating the data and services over multiple nodes, the system can support the failure of some of these nodes, without losing its ability to function correctly. Moreover, data and service replication are employed for load balancing reasons. In a typical distributed environment that collects or monitors data, useful data and services may be spread across multiple distributed nodes, but users or applications may wish to access that data through a central location (a proxy service). A common way to ensure centralized access to distributed data is by means of maintaining replicas of data objects of interest at a central location. However, when data collections are large or volatile, keeping replicas consistent with remote master copies poses a significant challenge due to the large communication cost incurred.

IV. Proposed Solution

There are several algorithms that have been used to find out the distances between the nodes in the network. Some of them are mentioned below: -

Estimating Hop Distance Between Arbitrary Host Pairs[12]
In this approach pairwise hop distances between nodes in the Internet is identified. This begins by deploying a landmark measurement infrastructure in the Internet, which uses active probes to establish accurate hop distances between all of its nodes. The measurement infrastructure is also enabled to passively monitor links and collect end host source addresses and TTLs from packets. It describe a multi - dimensional scaling algorithm that can be apply to the active and passive measurements that generates estimates of hop distances between all observed nodes (end host addresses and landmarks). The algorithm is designed to generate estimates even when all end host addresses have not been observed in all landmarks. The embedding algorithm is then enhanced to incorporate autonomous system information for the end hosts, which results in improved estimation performance.

A New Hierarchical Network Coordinate Algorithm Based on Community Structure[13]
In this paper, in order to reduce the impact of distance range on the prediction accuracy that short distance prediction suffers from high relative error, we propose a new hierarchical network coordinate algorithm based on community structure detection. The whole network is organized into N - level hierarchical community structure due to the host's geometric position distribution. Each host is assigned multiple coordinates corresponding to the hierarchical structure so that different sets of coordinates satisfy different ranges of distance prediction. Our algorithm is more applicable in constructing hierarchical coordinates more than two level that it requires no fixed infrastructure in the procedure of hierarchical structure organization.

By considering all the above mentioned algorithm, we can conclude that the hop distances between the nodes can be known.

Now, If the hop distances are known and if the security is to be imposed on to the hybrid P2P network, then after calculating the hop distances and if the hop distances are less (one or two hops) then 128 bit key length AES can be used and in case if the hop distances are more than two then 192 bit key length AES can be used and if the hop distances between the hosts are found more than 256 bit key length can be used, as it is obvious that the more is the distance between the source and the destination host, the more is the chance for the data insecurity.

Similarly once the hope distances are known and fault tolerance mechanism is to be used then based upon the hop distances the replicated peer can be placed to the maximum of hop distances of two to overcome the communication delay and thus the communication cost can be reduced.

V. Conclusion and Further Enhancement

We have just seen the advantages of knowing the hop distances in the network can be used in many ways to optimize the communications between the host node and the destination host node w.r.t to the security keys using AES, The lesser is the distance the lower is the key length, so that less time has to be consumed for the encryption as well as for the decryption and more is the distances higher level of the keys can be used. Same is the case with the fault tolerance techniques, where the identical peer can be placed to the nearby of the actual peer so that, if the actual peer fails due to some of the reason, the other replicated peer will be there to continue with the services. We can further enhance the above proposed idea using the Hidden Markov Model and can categories the hop distances into the three parts, i.e. the minimum hope distance, the average hope distance and the maximum hope distance between the hosts and based upon this automatically the keys length of the AES can be used for the communications and even it can be used to place the replicated data/peer to the nearby place of the actual peer which is currently giving the services. What we can do is that, the logic that has been mentioned above can be placed in the virtual node. Now the virtual node is the node which we can be placed between the clusters or an intermediate node between the clusters.

References

[1] Min Yang and Yuanyuan Yang, "*An Efficient Hybrid Peer - to - Peer System for Distributed Data Sharing*" IEEE Xplore November 2008.

[2] A. Davidson, *P2P File Sharing Privacy and Security*, Testimony before the House Committee on Government Reform Center for Democracy and Technology, May 2003.

[3] M. Castro, *Secure Routing for Structured P2P Overlay Networks*, In Proceedings of the 5th Usenix Symposium on Operating Systems Design and Implementation (OSDI 2002), Boston, Massachusetts, USA, December 2002.

[4] N. Daswani and H. Garcia - Molina, *Query - Flood DoS Attacks in Gnutella, In Proceedings of the 9th ACM Conference on Computer and Communications Security*, Washington DC, USA, November 2002, pp. 181 - 192.

[5] Alexandru Costan, Ciprian Dobre, Florin Pop, Catalin Leordeanu, Valentin Cristea, " *A Fault Tolerance Approach for Distributed Systems Using Monitoring Based Replication*"IEEE Xplore 2010.

[6] Ramesh Babu, George Abraham and Kiransinh Borasia, " *A Review On*

Securing Distributed Systems Using Symmetric Key Cryptography" International Journal of Advances in Science and Technology, Vol. 4, No.4, 2012.

[7] Ghemawat S, Gobioff H, Leung S. *The Google File System* In: the 19th ACM Symposium on Operating Systems Principles. Google, 2003.

[8] Preetinder Singh, *" AES Keys and Round Functions for Data Security*" International Journal of Computer Applications (0975 - 8887) Volume 39 - No.11, February 2012.

[9] Mayuri A. Mehta, *" Designing an Effective Dynamic Load Balancing Algorithm Considering Imperative Desing Issues in Distributed Systems "* International Conference on Communication Systems and Network Technologies, IEEE 2012.

[10] P. Werstein, H. Situ, and Z. Huang, *"Load Balancing in a Cluster Computing, "* Proc. 7th Int. Conf. Parallel and Distributed Computing, Applications and Technologies (PDCAT), IEEEComputer Society, Dec. 2006

[11] M. Beltran, A. Guzman, and J. L. Bosque, *"Dealing with Heterogeneity in Load Balancing Algorithms, "* Proc. 5th Int. Symp. Parallel and Distributed Computing, Jul. 2006, pp.123 - 132, doi: 10.1109/ISPDC.2006.17

[12] Brian Eriksson, Paul Barford, Robert Nowak, *" Estimating Hop Distance Between Arbitrary Host Pairs*" IEEE, 2009.

[13] Zilong Ye, Yabing Liu, Siguang Chen, *" A New Hierarchical Network Coordinate Algorithm Based on Community Structure*" International Conference on Computational Science and Engineering, IEEE 2009