Far East Journal of Electronics and Communications

Published Online: August 2016
http://dx.doi.org/10.17654/EC016030673
Volume 16, Number 3, 2016, Pages 673-683 **ISSN: 0973-7006**

IMPROVEMENT IN ROUTING TECHNIQUES IN P2P NETWORKS USING A CLOUD SERVICE INTERFACE WITH SECURE MULTIPARTY COMPUTATION

Anil Saroliya[1], Upendra Mishra[1] and Ajay Rana[2]

[1]Amity School of Engineering and Technology
Amity University
Jaipur, Rajasthan, India

[2]Amity University
Noida, Uttar Pradesh, India

Abstract

With the advent of latest technology, with every passing day there is an emergence of new domains at every step and the domain of networking is no exception. The interlinking of various computing devices otherwise distributed at different locations results in the P2P (peer to peer) network. Such devices uniquely contribute in the flow of communication and other data in terms of resource sharing viz. content or data (file) sharing even in the absence of any primary or intermediate authority. One of the most common and significant drawbacks in this type of architecture is security compromises amongst peers during the exchange of information. The pivot functioning of DHT-overlay routing protocol is to offer an approach to explore the resources in the peer to peer network. The present paper attempts to propose a routing algorithm to ensure the location of a specific node exploiting the available cloud service interface while achieving secure transaction from one node to another. The paper

Received: March 1, 2016; Revised: March 23, 2016; Accepted: May 18, 2016
Keywords and phrases: P2P networks, cloud computing, chord protocol, SMC (secure multiparty computation).

primarily examines and deals at length with chord protocol executing and utilizing cloud service, SMC (secure multiparty computation) technique as layered approach not only to enhance high security factors while transference of data but also to maintain transparency and reliability at all levels.

1. Introduction

The resources with DHT (distributed hash-table) oriented protocols (like chord, etc.) can be conveniently shared in distributed environment. Computing devices are physically apart and aloof with farthest distances, but logically they can be closer if a concept is introduced to abridge the same. The scientists are painstakingly working round the clock to explore a way in which the end-user will get the quick, complete and secure information that too without any overheads. P2P networks play a significant role as in such networks; information transfer is easily done by DHT overlay protocols. In DHT, several nodes are joined logically to develop unique cluster, every node keeps the small information of other nodes with respect to their location and resource-id. Consequently, searching and resource discovery of other nodes can be done efficiently. While resource searching and node identification is the primary task of DHT protocols, the other aspects like downloading or accumulating the resources (which remains the focal point of present paper) on nodes can fall under the responsibility of upper layers of P2P application and cloud infrastructure. The concept of cloud computing [1] is included here to reduce the side-effects of high churn rate of P2P networking in chord protocol [2]. With the help of cloud infrastructure (as shown in Figure 1), resource allocation can be performed through a secure layer, from such allocation not only possibility of false routing will be ruled out but also all resource related requests could easily be responded by cloud without facing any difficulty of whatsoever type.

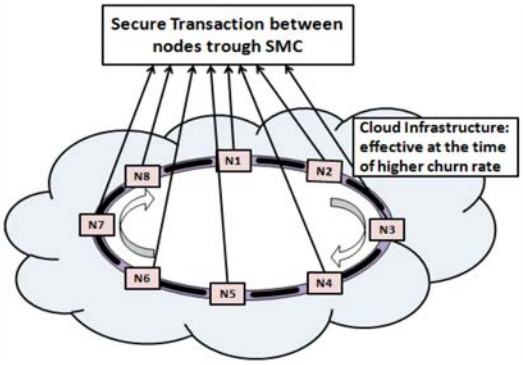


Figure 1. Conceptual architecture of cloud enabled SMC based chord-P2P network.

The concept of secure multiparty computation (used here) to provide secure communication channel enables the involved parties only to concentrate on outcome with no mutual information at any level. To achieve more clarity, we assume about a reliable third party, to which other parties provide their inputs where the former (third party) calculates the output and reverts the same to all the parties. In this, the responsibility of third party is provided to cloud infrastructure where the internal working of cloud remains unseen or unobserved by other parties via SMC [3] which gives opportunity to several parties; they can mutually carry out various comprehensive computations on their confidential data without compromising data security. It provides a platform for development of secure multiparty protocol in between end points [4].

The further categorization of this paper includes three sections which are as follows: Section 2 explains a brief working of DHT overlay supported P2P routing protocol while Section 3 elaborates two step solutions in detail; where the first part explains the involvement of cloud infrastructure and the later explains the functionality of SMC for secure transaction between two P2P nodes. The results after simulation are analyzed by comparing the different lookups between previous solutions (ordinary one) and present solution in Section 4.

2. Working of P2P Routing Protocol

In P2P network, every resource has a unique identifier named as a key, which is always associated with itself. With the help of a key, DHT overlay routing protocol (chord) can rapidly trace the node which is responsible for the related resource; characteristically in $O(\log n)$ hop counts if numbers of nodes are n in the P2P network. In this network, the chord protocol assigns numerical identifiers to both machine (nodes) and resources (keys) initially. The key identifier is obtained through hash function (after applying hash function on respective key); such hash function is used by every node of P2P system that returns m bit length of integers. Node identifier can be obtained by hashing IP address of respective node. After finding out both the identifiers, all the nodes and keys (related to respective nodes) are positioned in an identifier circle (ring) modulo $2m$. Each node holds the key value of resources (keys) with the respective key's identifier; such identifier can be equal to the identifiers of successive previous nodes which are not active on circular ring. Figure 2 illustrates the fundamental working of chord protocol. In this ring, the value of m (hash-bit) is 10. Here 11 nodes are active in the chord ring with 6 keys as indicated by the arrows. For searching the node which is liable for respective key, some path related information (routing information) is required to be stored on each node in a table called as "finger table".

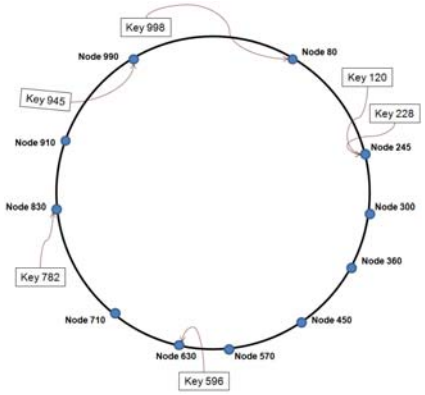


Figure 2. Process of keys mapping to nodes.

Finger table based on chord protocol keeps *m* entries with respective node-id (id) for a P2P node. Node entries are placed in table with $id + 2^i \pmod{2^k}$ shown in Figure 3. Finger table plays an important role for looking up the key in the chord-P2P network, all successors' positions are held by predecessor and searching of an appropriate resource can be done easily. In this P2P network, when a node leaves the network immediately all predecessors update their finger table accordingly; and resources which were held by the node which recently left, now its key is available to next updated successor as per the updated finger table.

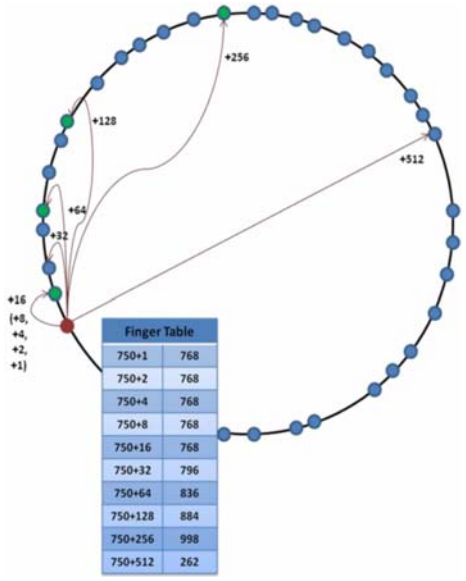


Figure 3. An example of a finger table.

In such type of flexible networks, the rate of recurrence by the node for moving out from network is very high, this rate is known as *churn rate*. When churning occurs in the network, then every node immediately works upon its routing path to update finger table quickly. This high rate of churn and frequent updating of finger table sometime hang the functionality of whole ring logically for a time being. So there must be a solution to handle such type of side effect of churn-rate. One of the solutions is introduced in following section by using the concept of cloud computing.

There are two types of main attacks which mostly tempered the structure of chord and introduced vulnerability [5]. First one in routing attack: intermediate node intentionally slumps the resource lookup-request and forwards the request to any node which leads infringement in routing policy of chord protocol. Another type of attack is: collusion, in collusion, intermediate node may behave as compromised node under the influence of any malicious node, so such node may reroute the request to any other malicious node and propagate the request in infinite loop. Due to such two types of attacks, chord enabled network demands some secured mechanism through which transaction can take place easily and eventually. In the next section, such type of attacks solution is also covered by using secure multipart computation (SMC).

3. Proposed Defense Mechanisms

Cloud computing is not a hardware product or software tool, but it is the collection of various services. Such services consumer can avail any time based on his/her demand [reference]. The effect of collusion can be reduced in chord ring by involvement of cloud service at some extent in which it does resource distribution for the nodes of P2P. Here an approach is proposed which provides a way to care about the resources from cloud. Nodes will only be responsible to take care about resource-ID. Through this approach, collusion can be easily avoided and resource packet will be reached to an appropriate destination.

3.1. Solution through cloud service

In this cloud enable P2P service, at a particular moment, if a node comes out of the P2P ring or network, successive nodes will take care about the resources id or key of the same node which left the network. Only resource id can be handled by successive nodes but the real resource will be guarded and held by cloud-infrastructure.

In a peer to peer network, a node asks for a particular resource (i.e., data file, audio file, video file, etc.) which does not exist due to inactive node

(currently not present in network). In the default working of chord protocol, such absent node resources are always taken care by successive nodes but there may be possibility of collusion by successive or next responsible node. Due to such collusion in this proposed approach, all the absent node resources will be taken care by cloud only (means additional option of cloud can be implant to store the copy of particular or any nodes resources). By using key (resource-id), the request can be automatically re-routed to cloud service infrastructure. Before resolving the resource query proper authentication of responsible (successive) node will be done, after completing this security check, resource query will be dealt by cloud infrastructure to the requested node through successive node. Cloud will always follow recursive approach so that overall originality and sanity of P2P network will be maintained. The role of cloud service in peer to peer network can be visualized through following figure.

3.2. Solution through SMC

With the cloud enabled solution, there is almost surety about particular resource. But always there may be possibility of man-in-middle attack or attack on availability during resource transaction between nodes. So the next step is the use of secure multiparty computation (SMC) to make secure transactions on the route. It can be done if the communicating parties share a common key to encrypt and decrypt data that will flow through this route. In this work, secure multiparty computation approach has been implemented so that the particular resource transaction can be done easily without involving any third party (after resolving resource request by cloud infrastructure) as a mediator.

In SMC, it has to be assumed that every node in the network maintains two parameters '*p*' and '*g*'. These both parameters are specified as public by default. Parameter '*g*' should always be less than '*p*' to satisfy the requirement of $n = g^k \bmod p$ [6], where '*k*' represents the private value of the particular node [SMC reference]. Such private data must be maintained by nodes which are trying to communicate using SMC.

Key generating method for building secure data transaction between node $N1$ and node $N21$:

I. Analysis on $N1$	**II. Analysis on $N21$**
Assume:	Assume:
Shared data between $N1$ and $N21 = p = 20$	Shared data between $N1$ and $N21 = p = 20$
Shared data between $N1$ and $N21 = g = 4$	Shared data between $N1$ and $N21 = g = 4$
Private data of $N1 = a = 8$	Private data of $N21 = b = 13$
$AA = g^a \bmod p$	$BB = g^b \bmod p$
$AA = 4^8 \bmod 20 = 16$	$BB = 4^{13} \bmod 20 = 4$
AA is send to $N21$	BB is send to $N1$
III. Secure multiparty key calculation at $N1$	**IV. Secure multiparty key calculation at $N21$**
On receiving BB from $N21$, $N1$ computes	On receiving AA from $N1$, $N21$ computes
$S_K = BB^a \bmod p = 4^8 \bmod 20 = 16$	$S_K = AA^b \bmod p = 16^{13} \bmod 20 = 16$

Now it has been clear with above calculations that same results are found at both ends. Such established two keys at both nodes can be utilized to put encryption and decryption processes on the packet which will transfer between $N1$ and $N21$. In such process, these nodes will never share their private data but the security will be maintained by getting same results at both sides. Such result can also work as session-key between two communicating parties. Similarly, such types of SMC calculations can be applied between other two nodes for secure data communication.

4. Results

In this paper, an enhanced chord enabled routing algorithm is proposed, named as AUR-chord. Following chart explains computations related to secure multiparty computation in AUR-chord:

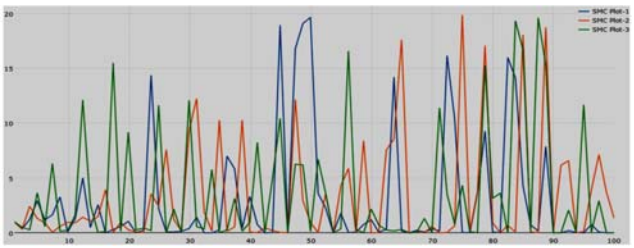
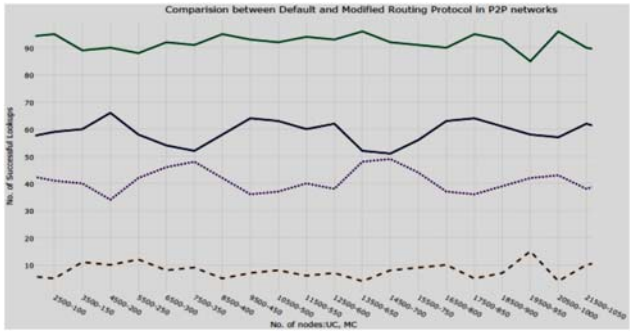
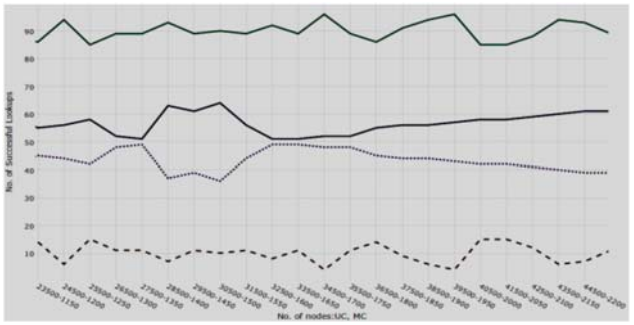


Figure 4. SMC in AUR-chord routing.

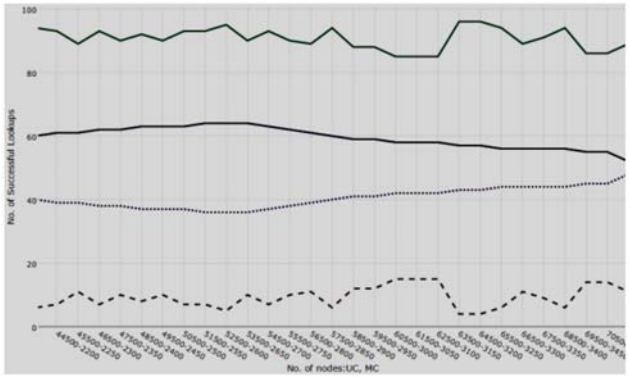
For enhanced routing in chord-P2P network, AUR-chord is implemented by creating a simulator using J2SE library and performed some experiments. In all experiments, the numbers of lookups are 100, out of which good and bad routing paths are achieved. After comparing AUR-chord with default one, following results have been identified in terms of Figure 5(a)-5(d). Such results are simulated through python programming environment.



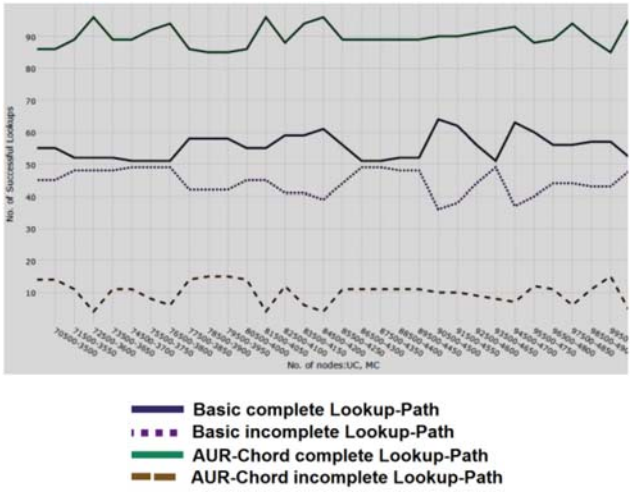
(a)



(b)



(c)



(d)

Figure 5(a)-5(d). Comparison between default and AUR-chord routing protocols in P2P networks.

5. Conclusion

Due to extensive churning in flexible chord-enabled P2P network, collusion can be easily introduced by upcoming nodes when requests of a particular resource are send by any predecessor node of the ring. In this paper, the presence of cloud infrastructure introduces an authentic layer for

fair distribution of resources. Through this paper, two layer approaches have been presented, first resource distribution through cloud and then the communication channel which will be established between two points remaining secure for data transaction using secure multiparty computation. SMC provides a way in which both the parties involved in communication learn nothing except results. Calculated results show the novel approach in structured P2P network which improves the performance of routing efficiently. Results explain that, new (proposed one) approach gives betterment in terms of higher side successful lookups as compared to older approach.

Acknowledgement

The authors thank the anonymous referees for their valuable suggestions which led to the improvement of the manuscript.

References

[1] P. Mathur and Nikhil Nishchal, Cloud computing: new challenge to the entire computer industry, Proc. IEEE PDGC'10, 2010, pp. 223-228.

[2] I. Stoica, Robert Morris, David Karger, M. Frans Kaashoek and Hari Balakrishnan, Chord: a scalable peer-to-peer lookup service for Internet applications, Proc. ACM SIGCOMM'01, 2001, pp. 149-160.

[3] W. Du and Mikhail J. Atallah, Secure multiparty computation problems and their applications: a review and open problems, Proc. ACM NSPW'01, 2001, pp. 13-22.

[4] D. K. Mishra, Purnima Trivedi and Samiksha Shukla, A glance at secure multiparty computation for privacy preserving data mining, International Journal on Computer Science and Engineering 1-3 (2009), 171-175.

[5] D. Wallach, A survey of peer-to-peer security issues, Int. Symposium on Software Security, Tokyo, Japan, 2002.

[6] W. Stallings, Cryptography and Network Security: Principles and Practice, Prentice Hall, 2010.