Efficient Secured Data Lookup and Multicast Protocols with Anonymity in RC-Based Two-level Hierarchical Structured P2P Network

Swathi Kaluvakuri*, Indranil Roy*, Koushik Maddali*, Bidyut Gupta*
Southern Illinois University Carbondale,Carbondale, IL USA

Narayan Debnath†
Eastern International University, VIETNAM

Abstract

In this paper, we have considered a recently reported 2-level non-DHT-based structured P2P network. It is an interest-based architecture. Residue Class (RC) based on modular arithmetic has been used to realize the overlay topology. Such an architecture has been the choice because it offers low latency in both inter or intra group communications. In the present work, we have proposed efficient ways to make these already existing communication protocols secured. In addition, we have extended these protocols further to include anonymity as well.

Key Words: Overlay multicast, residue class, interest - based, theorem, structured P2P networks, secured protocols, anonymity.

1 Introduction

Peer-to-Peer (P2P) overlay networks are widely used in distributed systems due to their ability to provide computational and data resource sharing capability in a scalable, self-organizing, distributed manner. P2P networks are classified into two classes: unstructured and structured ones. In unstructured systems [2] peers are organized into arbitrary topology. It takes the help of flooding for data look up. Problem arising due to frequent peer joining and leaving the system, also known as churn, is handled effectively in unstructured systems, however, it compromises with the efficiency of data query and the much-needed flexibility. In unstructured networks, lookups are not guaranteed. On the other hand, structured overlay networks provide deterministic bounds on data discovery. They provide scalable network overlays based on a distributed data structure which actually supports the deterministic behavior for data lookup. Recent trend in designing structured overlay architectures is the use of distributed hash tables (DHTs) [6, 9, 18]. Such overlay architectures can offer efficient, flexible, and robust service [6, 9, 11, 18-19].

However, maintaining DHTs is a complex task and needs substantial amount of effort to handle the problem of churn. So, the major challenge facing such architectures is how to reduce this amount of effort while still providing an efficient data query service. In this direction, there exist several important works, which have considered designing hybrid systems [5, 14, 16, 20]. These works attempt to include the advantages of both structured and unstructured architectures. However, these works have their own pros and cons [1].

*School of computing. E-mail: [swathi.kaluvakuri, indranil.roy, *S*koushik, bidyut]@siu.edu.
†School of Computing and Information Technology. E-mail: ndebnath@gmail.com

2 Preliminaries

Some of the preliminaries of this RC-based low diameter two level hierarchical structured P2P network [7-8, 12], have been considered here. In this section, we present a structured architecture for an interest-based peer-to-peer system. The following notations along with their interpretations will be used while we define the architecture.

Definition 1. We define a resource as a tuple $<Res_i, V>$, where Res_i denotes the type of a resource and $V$is the value of the resource. Note that a resource can have many values.

Definition 2. Let S be the set of all peers in a peer-to-peer system. Then $S = \{P^{Ri}\}$, $0 \le i \le n-1$, where P^{Ri} denotes the subset consisting of all peers with the same resource type *Resi*. and the number of distinct resource types present in the system is n. Also, for each subset P^{Ri}, we assume that H_i is the first peer among the peers in P^{Ri} to join the system. We call H_i as the group-head of group G_i formed by the peers in the subset P^{Ri}.

We now describe our proposed architecture suitable for interest-based peer-to-peer system. Generalization of the architecture is considered in [8].

We use the following notations along with their interpretations while we define the architecture.

2.1 Two Level Hierarchy

It is a two-level overlay architecture and at each level structured networks of peers exist. It is explained in detail below.

1) At level-1, we have a ring network consisting of the peers H_i ($0 \le i \le n-1$). The number of peers on the ring is n which is also the number of distinct resource types. This ring network is used for efficient data lookup and so we name it as transit ring

network.

2) At level-2, there are n numbers of completely connected networks (groups) of peers. Each such group, say G_i is formed by the peers of the subset P^{Ri}, $(0 \le i \le n-1)$, such that all peers (ϵ P^{Ri}) are directly connected (logically) to each other, resulting in the network diameter of 1. Each G_i is connected to the transit ring network via its group-head H_i.

3) Each peer on the transit ring network maintains a global resource table (GRT) that consists of n number of tuples. GRT contains one tuple per group and each tuple is of the form <Resource Type, Resource Code, Group Head Logical Address>, where Group Head Logical Address refers to the architecture. Also, Resource Code is the same as the group-head logical address.

4) Any communication between a peer $G_{x,i} \in$ group G_x and $G_{y,j} \in$ group G_y takes place only through the corresponding group heads H_x and H_y.

The proposed architecture is shown in Figure 1.

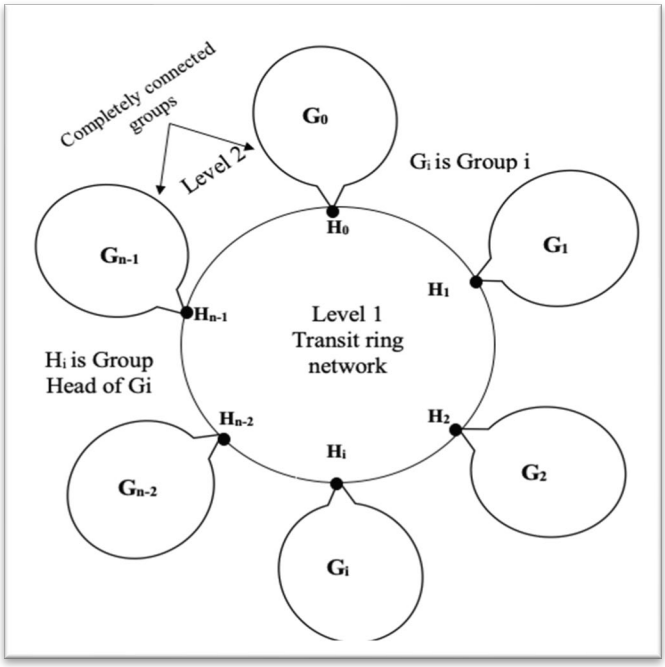

Figure 1: A two-level RC based structured P2P architecture with n distinct resource types

2.2 Relevant Properties of Modular Arithmetic

Consider the set S_n of nonnegative integers less than n, given as $S_n = \{0, 1, 2, \ldots (n-1)\}$. This is referred to as the set of residues, or residue classes (mod n). That is, each integer in S_n represents a residue class (RC). These residue classes can be labelled as [0], [1], [2], …, [n – 1], where [r] = {a: a is an integer, $a \equiv r \pmod n$}.

For example, for n = 3, the classes are:

$[0] = \{\ldots, -6, -3, 0, 3, 6, \ldots\}$
$[1] = \{\ldots, -5, -2, 1, 4, 7, \ldots\}$
$[2] = \{\ldots, -4, -1, 2, 5, 8, \ldots\}$

Thus, any class r (mod n) of S_n can be written as follows:

$[r] = \{\ldots, (r - 2n), (r - n), r, (r + n), (r +2 n), \ldots, (r + (j-1).\ n), (r + j.n), (r + (j+1).n), \ldots..\}$

A few relevant properties of residue class are stated below.

Lemma 1. Any two numbers of any class r of S_n are mutually congruent.

2.3 Assignments of Overlay Addresses

Assume that in an interest-based P2P system there are n distinct resource types. Note that n can be set to an extremely large value *a priori* to accommodate a large number of distinct resource types. Consider the set of all peers in the system given as $S = \{P^{Ri}\}$, $0 \le i \le n-1$. Also, as mentioned earlier, for each subset P^{Ri} (i.e. group G_i) peer H_i is the first peer with resource type R_i to join the system.

The assignment of logical addresses to the peers at the two levels and the resources happen as explained in [7-8, 12].

Remark 1. GRT remains sorted with respect to the logical addresses of the group-heads.

Definition 3. Two peers H_i and H_j on the ring network are logically linked together if $(i + 1) \bmod n = j$.

Remark 2. The last group-head H_{n-1} and the first group-head P_0 are neighbors based on Definition 3. It justifies that the transit network is a ring.

Definition 4. Two peers of a group G_r are logically linked together if their assigned logical addresses are mutually congruent.

Lemma 2. Diameter of the transit ring network is n/2.
Lemma 3. Each group G_r forms a complete graph.

2.4 Salient Features of Overlay Architecture

We summarize the salient features of this architecture.

1) It is a hierarchical overlay network architecture consisting of two levels; at each level the network is a structured one.

2) Use of modular arithmetic allows a group-head address to be identical to the resource type owned by the group. We will show in the following section the benefit of this idea from the viewpoint of achieving reasonably very low search latency.

3) Number of peers on the ring is equal to the number of distinct resource types, unlike in existing distributed hash table-based works some of which use a ring network at the heart of their proposed architecture [11].

4) The transit ring network has the diameter of n/2. Note that in general in any P2P network, the total number of peers N >> n.

5) Each overlay network at level 2 is completely connected. That is, in graph theoretic term it is a complete graph consisting of the peers in the group. So, its diameter is just 1. Because of this smallest possible diameter (in terms of number of overlay hops) the architecture offers minimum search latency inside a group.

2.5 Our Contribution

In this paper, we have considered interest based P2P systems [20-21]. We have considered designing secured protocols for Inter and Intra lookup algorithms. The concepts of symmetric key, asymmetric key cryptography with public and private keys have been used. We have also considered making the capacity-constrained multicast algorithms more efficient with security both inside a group and for the two-level architecture. In addition, we have also considered anonymity. In section III we present the secured data lookup algorithms and in section IV, we have present multicast algorithms with both security and anonymity.

3 Data Lookup Algorithms with Security

Cryptography is the research and implementation of encrypted communication techniques. It is concerned with the creation and analysis of protocols that prevent malicious third parties from accessing information exchanged between two organizations, thereby adhering to various aspects of information security.

A situation in which a message or data exchanged between two parties cannot be accessed by an adversary is referred to as secure communication. In cryptography, an adversary is a malicious party that attempts to retrieve useful information or data by breaching information security principles.

Cryptographic algorithms are used to achieve stability in peer-to-peer networks in terms of authentication and confidentiality. Secret key cryptographic algorithms and public key cryptographic algorithms are the two types of cryptographic algorithms that are most used. Since the same key is used for encryption and decryption and is shared by all parties concerned, secret key cryptographic algorithms are also known as symmetric key algorithms. Asymmetric key algorithms, on the other hand, are also known as public key cryptographic algorithms. A pair of keys, one for encryption and the other for decryption, are used in this form. One of the keys, known as the public key, is made public, while the other, known as the private key, is kept private. Only the pair's secret key will decrypt a message encrypted with a public key. Similarly, a message encrypted with a private key can only be decrypted by the pair's public key [13]. Data lookup algorithms [7, 12] both Inter and Intra are presented in this section with the concept of security. Cryptographic functions and their applications in 2 level RC based architecture is explained in Figure 2.

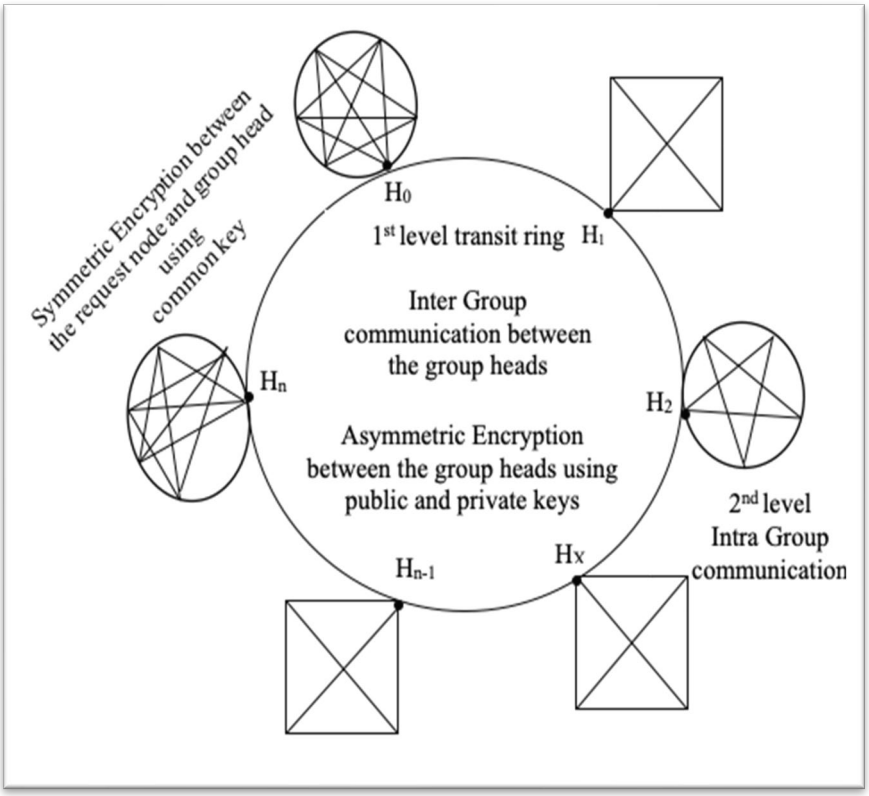

Figure 2: Cryptographic functions and their applications in 2 level architecture

3.1 Intra Group Lookup Algorithm with Security in RC based Architecture

In this case the resource lookup happens within the group, i.e., the resource type is the same for both the parties but the value is different. So, we use the concept of symmetric key cryptography where the same cryptographic key is used for encoding the message (request) by requesting peer and decoding the message (request) by group head. The algorithm is explained as follows.

Let us assume that in a group G_x , a peer $G_{x,i}$ with resource $<Res_x, V_i>$ is looking for a resource $<Res_x, V_j>$. *Let* $SyKey_{x,i}$ is the common/symmetric key shared by the requesting node $G_{x,i}$ and the corresponding group head H_x of the same group G_x as explained in Figure 3.

Secured Intra-Lookup Algorithm

1. Request node $G_{x,i}$ will encrypt the message $<Res_x, V_j>$ with symmetric key $SyKey_{x,i}$ and sends it group head H_x through a unicast message.

   // To make it clear, this symmetric key information is known only to the requestor and the group head so other nodes in the same interest group will not be able to decrypt the request.

2. The group head H_x will then decrypt the encrypted request with the symmetric key $SyKey_{x,i}$
3. Later, this request $<Res_x, V_j>$ will be broadcasted in the interest group G_x by the group head H_x
4. **If** a node $G_{x,j}$ *in* group G_x has the requested resource $<Res_x, V_j>$

   a. it encrypts the resource $<Res_x, V_j>$ with symmetric key $SyKy_{x,j}$ and unicasts it to the group head H_x
   b. The group head H_x will use the symmetric key $SyKy_{x,j}$ and decrypts the response from $G_{x,j}$
   c. H_x will now encrypt the response $<Res_x, V_j>$ with the symmetric key $SyKey_{x,i}$ and unicasts it to the requesting node $G_{x,i}$
   d. Finally, $G_{x,i}$ will decrypt the response using the symmetric key $SyKey_{x,i}$

else search for $<Res_x, V_j>$ fails

Figure 3: Intra Group Lookup Algorithm with Security in RC based architecture

3.2 Inter Group Lookup Algorithm with Security in RC based Architecture

When it comes to Inter group, the communication happens between the nodes from two different interest-based groups, so here comes the concept of public and private keys. Hence the asymmetric key security. In our secured RC based architecture, any sort of communication between the peers $G_{x,i} \in$ group G_x and $G_{y,j} \in$ group G_y takes place only through the corresponding group heads H_x and H_y.

The following notations are used to denote the public and private keys of the requesting and responding group heads.

- Pbl_x and Pvt_x to denote respectively the public and private keys of group-head H_x of group G_x.
- Pbl_y and Pvt_y to denote respectively the public and private keys of group-head H_y of group G_y.

Without any loss of generality, let a peer $G_{x,i} \in$ group G_x requests for a resource $<Res_y, V_j>$. Peer $G_{x,i}$ and group head H_x is aware of the fact that that $Res_y \notin$ group G_x. The secured inter-lookup algorithm is explained in Figure 4.

Secured Inter-Lookup Algorithm

1. Request node $G_{x,i}$ encrypts the request $<Res_y, V_j>$ using the common key $SyKey_{x,i}$ and unicasts it to the group head H_x
2. The group head H_x will then decrypt the encrypted request with the symmetric key $SyKey_{x,i}$. Because the group head H_x is aware of the fact that that $Res_y \notin$ group G_x, it finds the Group head address of H_y along with its public key Pbl_y from the GRT table.

   // address code of H_y *= resource code of* $Res_y = y$ *//*

3. H_x encrypts the message with Pbl_y and computes $|x - y| = h$
4. **if** $h > n / 2$ (where n is the number of distinct resource types)

   H_x forwards the request along with the IP address of the request node $G_{x,i}$ to its immediate predecessor H_{x-1}

   else H_x forwards the request along with the IP address of the request node $G_{x,i}$ to its immediate successor H_{x+1}

   // *Looking for minimum no. of hops*

   end

5. Each intermediate group-head H_k forwards the encrypted request until $H_k = H_y$

// In the worst case it will take around n/2 hops

6. Now H_y will decrypt the message using its private key Pvt_y
7. **if** H_y itself has the resource $<Res_y, V_j>$

   H_y encrypts the message with the public key Pbl_x of H_x and unicasts it to H_x

   else

   H_y broadcasts the request for $<Res_y, V_j>$ in group G_y

if ∃ $G_{y,j}$ (ϵG_y) which has the resource <Res_y, V_j>

- $G_{y,j}$ encrypts the message with symmetric key $SyKey_{y,j}$ and unicasts it to H_y.
- H_y decrypts the message with $SyKey_{y,j}$
- H_y encrypts the decrypted message with the public key Pbl_x of H_x and sends it to H_x
- H_x decrypts the message with its own private key Pvt_x
- Now, H_x encrypts the message <Res_y, V_j> with $SyKey_{x,i}$ and sends it to the requesting peer $G_{x,i}$
- $G_{x,i}$ will then decrypt the received message using the symmetric key $SyKey_{x,i}$

else

H_y unicasts 'search failed message' to H_x

end

end

Figure 4: Intra Group Lookup Algorithm with Security in RC based architecture

4 Multicast Algorithms with Anonymity and Security

The basic multicast algorithms of RC based architecture presented in [12] have been enhanced in this section with the concepts of security and anonymity. In [12] we have considered designing a highly efficient capacity-constrained overlay multicast protocol. Our architecture is a 2-level one. Number of nodes (group-heads) *n* on the level-1 ring is just the number of distinct resource types and in any group (cluster) at level 2 there can be any number of nodes. Note that the number of distinct resources *n* is much smaller than the total number of nodes *N* on the ring in [4]. It has inspired us to use some idea from [4], especially transforming the multicast problem to a broadcast one and appropriately augmenting it with ours to design a highly efficient any source capacity-constrained multicast protocol suitable for the RC-based architecture with much less hop and communication complexities compared to the work in [4]. The multicast algorithm with anonymity where $c^s_x \geq n_r$ is explained in Figure 5.

4.1 Multicast Algorithm [12] with Anonymity where capacity of group head ≥ #groupheads

Scenario 1: $c^s_x \geq n_r$ (with Anonymity)

1. Source peer $G_{x,i}$ unicasts *mcast_msg* to its group head H_x.
2. H_x gathers all ip_addresses of fellow groups heads from the GRT. H_x replaces the ip_address of $G_{x,i}$ to its own in the *mcast_msg* it received and unicasts to all fellow group heads participating in the multicast.
3. **If** a receiver group head is also a multicast group member,
   a. It makes a copy of the *mcast_msg* and keeps it for itself.
   b. Replaces the ip_address of H_x in the *mcast_msg* to itself and unicasts to each of its members.

   else

   Replaces the ip_address of H_x in the *mcast_msg* to itself and unicasts to each of its members.

   end

Figure 5: Multicast protocol with anonymity where capacity of the group head (c^s_x) ≥ number of receiver group-heads (n_r)

4.2 Multicast Algorithm [12] with Security and Anonymity where capacity of group head ≥ #groupheads

The multicast algorithm where $c^s_x \geq n_r$ considering the concepts of anonymity and security is explained in Figure 6.

Scenario 2: $c^s_x \geq n_r$ (with Anonymity and Security)

1. Source peer $G_{x,i}$ encrypts the message *mcast_msg* using the symmetric key $SyKey_{x,i}$ and unicasts it to the group head H_x.
2. Group head H_x decrypts the received *mcast_msg* using the symmetric key $SyKey_{x,i}$ and then replaces the ip_address of the $G_{x,i}$ to its own. // Anonymity
   // Note: GRT is modified in this scenario. public key of each.
3. H_x then gathers ip_addresses and the corresponding public keys of fellow group heads from the GRT.
4. Then, H_x will encrypt the modified *mcast_msg* with the public keys of the respective target/multicast group heads.
5. H_x will now unicast the encrypted *mcast_msg* to the target group head and repeats the same for all other group heads participating in the multicast.
6. When the message is received, each receiver group head decrypts the received *mcast_msg* using their respective private keys.
7. **If** the receiver group head is also a multicast group member,
   a. It makes a copy of the *mcast_msg* and keeps it for itself.
   b. Replaces the ip_address of H_x to its own, encrypts the message using symmetric key $SyKey_{a,b}$ (where a is the group head number and b is the number of group member) and unicasts it to each receiver.

   Else

   Replaces the ip_address of H_x to its own, encrypts the

message using symmetric key $SyKey_{a,b}$ (where a is the group head number and b is the number of group member) and unicasts it to each receiver.

Figure 6: Secured Multicast protocol with Anonymity where capacity of the group head (c^s_x) ≥ number of receiver group-heads (n_r)

Example 1:

Let us consider a scenario where $c^s_x \geq n_r$

group heads (n_r) = 7 (H_0 to H_6)
Assume that the capacity of each group head (c^s_x) = 9
Source Peer is $G_{5,12}$

In the example, Figure 7, source peer $G_{5,12}$ encrypts the *mcast_msg* using the common shared key $SyKey_{5,12}$ to the head of the group H_5. When group head H_5 receives the message, it decrypts *mcast_msg* using the common key $SyKey_{5,12}$ and then replaces the ip_address of the $G_{5,12}$ with its own address.

Now, H_5 gets the necessary information (ip_addresses and their respective public keys) of the target multicast group say

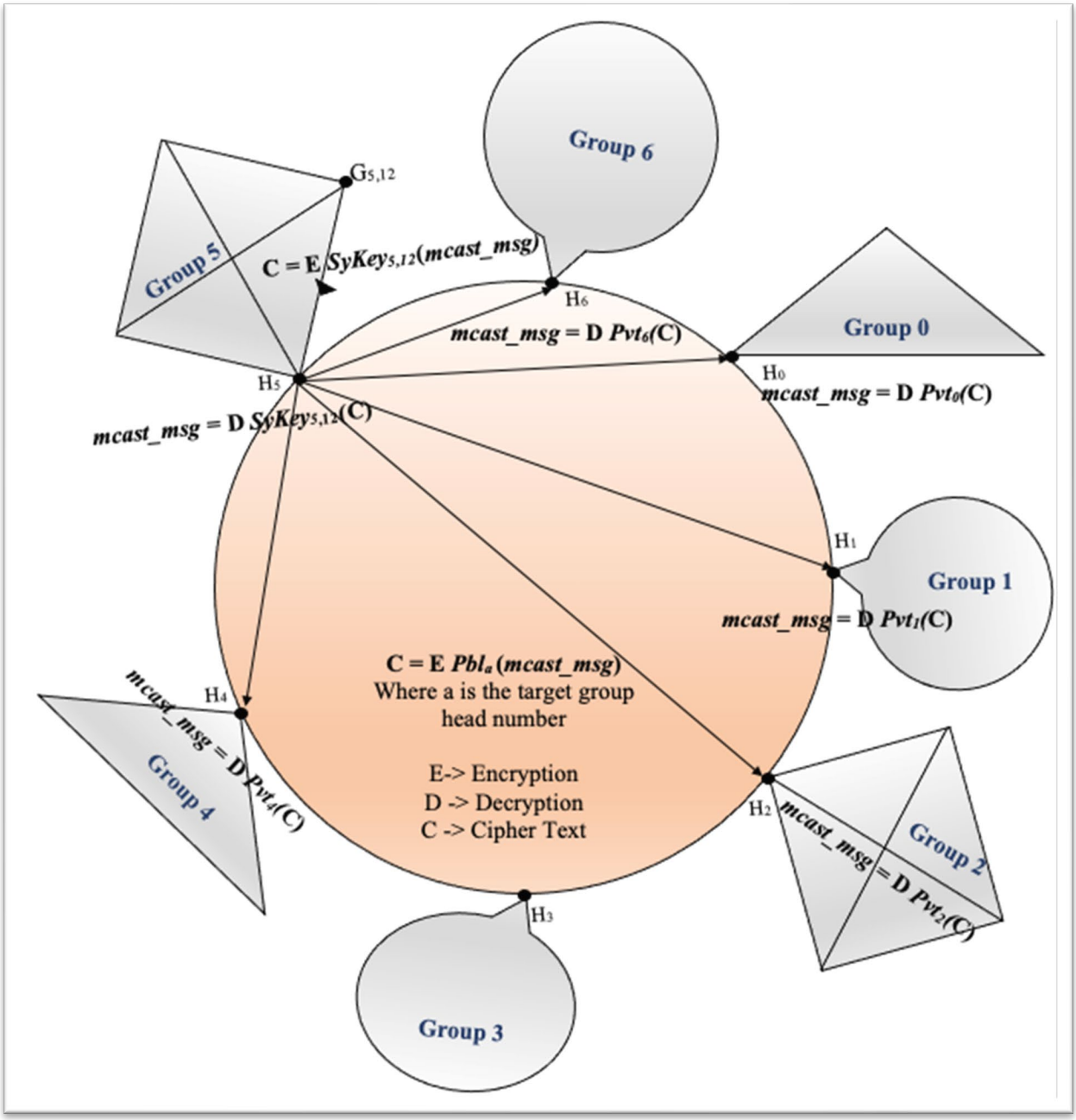

Figure 7: Example of secured multicast protocol when $c^s_x \geq n_r$

H_0, H_1, H_2, H_4 and H_6 from the global resource table (GRT) and then modifies *mcast_msg* by encrypting it with the publickeys Pbl_0, Pbl_1, Pbl_2, Pbl_4 and Pbl_6 respectively and unicasts the messages. On the receiving end H_0, H_1, H_2, H_4 and H_6 decrypts the *mcast_msg using the private keys* Pvt_0, Pvt_1, Pvt_2, Pvt_4 and Pvt_6.

Each head of the target group unicasts the message in two scenarios. For example,

- Group head H_0 on receiving the *mcast_msg* from H_5. makes a copy before encrypting it with symmetric keys and unicasting the message to each receiver in its group.
- Group head H_1 on receiving the *mcast_msg* from H_5. encrypts the received message using the symmetric key and unicasts it to each receiver in its group without making a copy for itself because it is not a multicast group member.

The multicast algorithm with anonymity where $c^s_x < n_r$ is explained in Figure 8, with anonymity and security is explained in Figure 9.

4.3 Multicast Algorithm [12] with Anonymity where capacity of group head < #groupheads

Scenario 3: $c^s_x < n_r$ (with Anonymity)

1. Source peer $G_{x,i}$ unicasts *mcast_msg* to its group head H_x.
2. H_x replaces the ip address of $G_{x,i}$ with its own.
3. H_x then gathers ip_addresses of all group heads participating in the multicast from the GRT.
4. H_x then randomly selects the group heads and unicasts the *mcast_msg* based on its capacity.
5. Every receiver group head changes the ip_address of H_x present in the received mcast_msg to its own and forwards it to its successor group head on the ring.
6. Additionally, if the receiver group head is a multicast member, it saves a copy for itself.
7. Each receiver group head also sends the *mcast_msg* to all of its members.
8. Message propagation among successor group heads continues around the 1st level circle.
9. A receiver group head drops the received *mcast_msg* if it has received it already from a different source.

Figure 8: Multicast protocol with anonymity where capacity of the group head (c^s_x) < number of receiver group-heads (n_r)

4.4 Multicast Algorithm [12] with Security and Anonymity where capacity of group head < #groupheads

Scenario 4: $c^s_x < n_r$ (with Security and Anonymity)

1. Source peer $G_{x,i}$ encrypts the message *mcast_msg* using the symmetric key $SyKey_{x,i}$ and unicasts it to the group head H_x.
2. Group head H_x decrypts the received *mcast_msg* using the symmetric key $SyKey_{x,i}$ and then replaces the ip_address of the $G_{x,i}$ to its own. *// Anonymity and Step 1 & 2 is similar to scenario 1*
3. H_x then gathers ip_addresses and the corresponding public keys of fellow group heads from the GRT.
4. H_x randomly selects those many groups heads equal to its capacity/degree.
5. H_x will also retrive it's successor's ip address and public key from the GRT table.
6. It encrypts the modified *mcast_msg* with the public key of the selected group head Pbl_a where a is the number of the group head on the transit ring and unicasts it.

   // address code of H_a = resource code of Res_a = a //

7. H_x will now unicast the encrypted *mcast_msg* to the selected group heads as well as its successor

   // one logical hop to each group head

8. **If** receiver group head receives the *mcast_msg* for the first time (unique),

   a. Each receiver group head decrypts the received *mcast_msg* using private key Pvt_a.

   If the receiver group head is also a multicast group member,

   i. It makes a copy of the *mcast_msg* and keeps it for itself.
   ii. Replaces the ip_address of H_x to its own, encrypts the message using symmetric key $SyKey_{a,b}$ (where a is the group number and b is the number of group members) and unicasts it to members.

   else

   Replaces the ip_address of H_x to its own, encrypts the message using symmetric key $SyKey_{a,b\,b}$ (where a is the group number and b is the number of group member) and unicasts it to members.

   end

   b. It then replaces the ip_address of H_x to its own; acquires ip_address and public key of successor group head; encrypts the modified message using the acquired public key Pbl_s where s is the successor group head and forwards it.
   c. Message propagation continues similarly in the 1st level ring until the message reaches all the group heads on level 1 transit ring

   else

Receiver group head drops the duplicate message

Figure 9: Secured multicast protocol with anonymity where capacity of the group head (c^s_x) < number of receiver group-heads (n_r)

Example 2:

Let us consider a scenario where $c^s_x < n_r$
group heads (n_r) = 7 (H_0 to H_6)
Assume that the Capacity of each group head (c^s_x) = 2
Source Peer is $G_{5,12}$

In this example, Figure 10, source peer $G_{5,12}$ encrypts the *mcast_msg* using the common shared key $SyKey_{5,12}$ to the head of the group H_5. When group head H_5 receives the message, it decrypts *mcast_msg* using the common key $SyKey_{5,12}$ and then replaces the ip_address of the $G_{5,12}$ with its own address (same as Example 1).

H_5 selects any 2 group heads in random (say H_1, H_4) and encrypts the *mcast_msg* with the public keys Pbl_1, Pbl_4, respectively and unicasts the message. The private keys Pvt_1, Pvt_4 respectively are used at the receiving end by the group heads to decode/decrypt the message. H_5 also unicasts the message to its successor H_6 as explained above.

Each group head that receives the encrypted message will unicast the message in the following scenarios.

- Group head H_1 on receiving the *mcast_msg* from H_5, makes a copy before encrypting it with symmetric keys and unicasting the message to each receiver in its group. H_1 also encrypts the message and forwards it to the successor

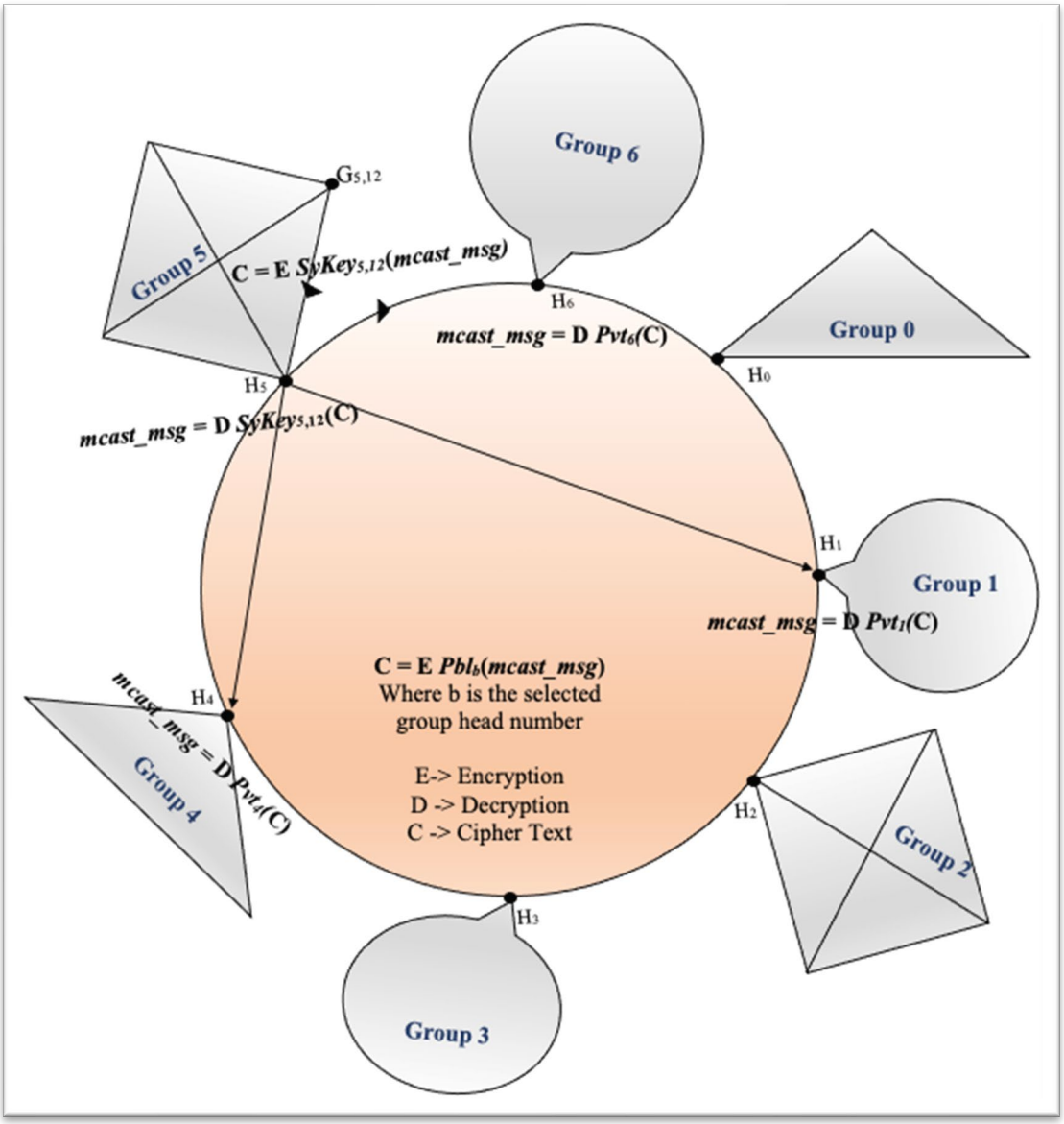

Figure 10: Example of Secured Multicast protocol when $c^s_x < n_r$

on the transit ring.

- Group head H_4 on receiving the *mcast_msg* from H_5, encrypts the received message using the symmetric key and unicasts it to each receiver in its group without making a copy for itself because it is not a multicast group member.

5 Conclusion

In this paper, we have considered a 2-level non DHT-based P2P architecture. This interest-based architecture has been the choice because

1. We have shown earlier [8] its superiority from the viewpoint of search latency of the data lookup protocols compared to those in some very prominent DHT-based contributions [15, 17, 22] and
2. Its superiority over several existing interest-based architectures [1, 3, 6, 9-10, 18]. In this paper, we have incorporated in a very effective way both security and anonymity in both inter and intra-group communication protocols which have appeared in [8].

Future work is directed at designing secured communication protocols for P2P federation built with multiple RC-based P2P components.

References

[1] L. Badis, M. Amad, D. Aîssani, K. Bedjguelal and A. Benkerrou, "ROUTIL: P2P Routing Protocol Based on Interest Links," 2016 International Conference on Advanced Aspects of Software Engineering(ICAASE), Constantine, pp. 1-5, 2016, doi: 10.1109/ICAASE.2016.7843852

[2] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making Gnutella-like P2P Systems Scalable," *Proc. ACM SIGCOMM*, Karlsruhe, Germany, pp. 407-418, August 25-29 2003.

[3] Wen-Tsuen Chen, Chi-Hong Chao and Jeng-Long Chiang, "An Interested-based Architecture for Peer-to-Peer Network Systems," 20th International Conference on Advanced Information Networking and Applications - Volume 1 (AINA'06) , Vienna, 2006, pp. 707-712, doi: 10.1109/AINA.2006.93

[4] Shiping Chen, Baile Shi, Shigang Chen, and Ye Xia, "ACOM: Any-Source Capacity-Constrained Overlay Multicast in Non-DHT P2P Networks," *IEEE Tr. Parallel and Distributed Systems*, 18(9):1188-1201, Sep. 2007.

[5] P. Ganesan, Q. Sun, and H. Garcia-Molina, "Yappers: A Peer-to-Peer Lookup Service over Arbitrary Topology," *Proc. IEEE Infocom 2003*, San Francisco, USA, 2:1250-1260, March 30 - April 1 2003.

[6] M. Hai and Y. Tu, "A P2P E-Commerce Model Based on Interest Community," 2010 International Conference on Management of e-Commerce and e-Government, Chengdu, pp. 362-365, 2010, doi: 10.1109/ICMeCG.2010.80

[7] Swathi Kaluvakuri, Koushik Maddali, Bidyut Gupta and Narayan Debnath, "Design of RC Based Low Diameter Hierarchical Structured P2P Network Architecture," EMENA-ISTL, 2019; *LAIS (Learning and Analytics in Intelligent Systems)*, Springer, 7:312-320, 2020.

[8] Swathi Kaluvakuri, Koushik Maddali, Nick Rahimi, Bidyut Gupta and Narayan Debnath, "Generalization of RC-Based Low Diameter Hierarchical Structured P2P Network Architecture", *International Journal of Computers and Their Applications (IJCA)*, 27(2):77-83, June 2020.

[9] Khambatti, Mujtaba & Ryu, Kyung and Dasgupta, Partha, "Structuring Peer-to-Peer Networks Using Interest-Based Communities," Lecture Notes in Computer Science, 1st International Workshop, DBISP2P 2003, Berlin, September 2003.

[10] S. K. A. Khan and L. N. Tokarchuk, "Interest-Based Self Organization in Group-Structured P2P Networks," 2009 6th IEEE Consumer Communications and Networking Conference, Las Vegas, NV, pp. 1-5, 2009, doi: 10.1109/CCNC.2009.4784959.

[11] D. Korzun and A. Gurtov, "Hierarchical Architectures in Structured Peer-to-Peer Overlay Networks," Peer-to-Peer Networking and Applications, Springer, pp. 1-37, March 2013.

[12] Koushik Maddali, Banafsheh Rekabdar, Swathi Kaluvakuri and Bidyut Gupta, "Efficient Capacity-Constrained Multicast in RC based P2P Networks," EPiC Series in Computing, *CAINE*, 63:121-129, September 2019

[13] Koushik Maddali, Swathi Kaluvakuri, Bidyut Gupta and Narayan Debnath, "On Designing Secured Communication Protocols along with Anonymity for CRT based Structured P2P Network Architecture," EPiC Series in Computing, *CAINE*, October 2020 (accepted).

[14] Z. Peng, Z. Duan, J. Jun Qi, Y. Cao, and E. Lv, "HP2P: A Hybrid Hierarchical P2P Network," *Proc. Intl. Conf. Digital Society*, pp. 18-28, 2007.

[15] A. Rowstron and P. Druschel, "Pastry: Scalable, Distributed Object Location and Routing for Large Scale Peer-to-Peer Systems," *Proc. FIP/ACM Intl. Conf. Distributed Systems Platforms (Middleware)*, pp. 329-350, 2001.

[16] K. Shuang, P Zhang, and S. Su, "Comb: A Resilient and Efficient Two-Hop Lookup Service for Distributed Communication System," *Security and Communication Networks*, 8(10):1890-1903, 2015.

[17] R. I. Stocia, R. Morris, D. Liben-Nowell, D. R. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications," *IEEE/ACM Tran. Networking*, 11(1):17-32, Feb. 2003.

[18] Z. Tu, W. Jiang and J. Jia, "Hierarchical Hybrid DVE-P2P Networking Based on Interests Clustering," 2017 International Conference on Virtual Reality and Visualization (ICVRV), Zhengzhou, China, pp. 378-381, 2017, doi: 10.1109/ICVRV.2017.00087.

[19] M. Xu, S. Zhou, and J. Guan, "A New and Effective Hierarchical Overlay Structure for Peer-to-Peer Networks," *Computer Communications*, 34:862-874, 2011.
[20] M. Yang and Y. Yang, "An Efficient Hybrid Peer-to-Peer System for Distributed Data Sharing," *IEEE Trans. Computers*, 59(9)1158-1171, Sep. 2010.
[21] R. Zhang and Y.C. Hu, "Assisted Peer–to-Peer Search with Partial Indexing," *IEEE Trans. Parallel and Distributed Systems*, 18(8):1146-1158, 2007.
[22] B. Y. Zhao, L. Huang, S. C. Rhea, J. Stribling, A. Zoseph, and J. D. Kubiatowicz, "Tapestry: A Global-Scale Overlay for Rapid Service Deployment," *IEEE J-SAC*, 22(1):41-53, Jan. 2004.

Swathi Kaluvakuri (photo not available) is a Ph.D. candidate from Southern Illinois University Carbondale – School of Computing. She graduated from Jawaharlal Nehru Technological Unversity with a Bachelor of Technology degree in Computer Science major. She holds keen interest in the areas of Peer to Peer Networking and BlockChain and worked as a Software Engineer, Technical Product Support as IBM AS400 developer for NetCracker Pvt Ltd from 2012-2014.

Indranil Roy (photo not available) is currently a PhD student in Computer Science department of Southern Illinois University, Carbondale. He has completed his B.E in Electronics and Communication from RCCIIT, Kolkata, in the year 2016. He received his M.S degree in Computer Science from Southern Illinois University, Carbondale in 2018. His main research interests include Blockchain along with interest-based p2p architecture.

Koushik Maddali (photo not available) is a Ph.D. candidate in Department of Computer Science at Southern Illinois University Carbondale. He received his MS from the same university and his BS from Jawaharlal Nehru Technological University, India. His research interests include Peer to Peer Networking, BlockChain and worked on a Virtual Terminal project of Cisco from 2017-2018.

Bidyut Gupta (photo not available) received his M. Tech. degree in Electronics Engineering and Ph.D. degree in Computer Science from Calcutta University, Calcutta, India. At present, he is a professor at the School of Computing (formerly Computer Science Department), Southern Illinois University, Carbondale, Illinois, USA. His current research interest includes design of architecture and communication protocols for structured peer-to-peer overlay networks, security in overlay networks, and block chain. He is a senior member of IEEE and ISCA.

Narayan Debnath (photo not available) earned a Doctor of Science (D.Sc.) degree in Computer Science and also a Doctor of Philosophy (Ph.D.) degree in Physics. Narayan C. Debnath is currently the Founding Dean of the School of Computing and Information Technology at Eastern International University, Vietnam. He is also serving as the Head of the Department of Software Engineering at Eastern International University, Vietnam. Dr. Debnath has been the Director of the International Society for Computers and their Applications (ISCA) since 2014. Formerly, Dr. Debnath served as a Full Professor of Computer Science at Winona State University, Minnesota, USA for 28 years (1989-2017). Dr. Debnath has been an active member of the ACM, IEEE Computer Society, Arab Computer Society, and a senior member of the ISCA.