Efficient and Secured Data Lookup Protocol using Public-Key and Digital Signature Authentication in RC-Based Hierarchical Structured P2P Network

Anjila Neupane*, Reshmi Mitra *, Indranil Roy *
Southeast Missouri State University, Cape Girardeau, MO 63701, USA

Bidyut Gupta †
Southern Illinois University, Carbondale, IL 62901, USA

Narayan Debnath ‡
Eastern International University, VIETNAM

Abstract

Peer-to-peer (P2P) networks are extremely vulnerable to network security attacks due to their low budget and limited resource capabilities. In this study, we are presenting public key cryptography and digital signature-based data lookup protocol that are efficient and secured in an RC-based hierarchical structured P2P network. Utilizing RC, a modular arithmetic-based residue class, the overlay topology has been achieved. This design was chosen because it allows for minimal latency in both intra and inter-group communications. In the present study, we provide efficient schemes for public-key cryptographic security and digital signature-based authentication for existing communication protocols.

Key Words: Public-key cryptography; digital signature; P2P networks ; interest-based network formation

1 Introduction

Due to their capacity to offer computational and data resource sharing in a scalable, self-organizing, distributed manner, peer-to-peer (P2P) overlay networks are widely used in distributed systems. P2P networks are divided into two categories: unstructured networks and structured networks. Peers in unstructured systems [2] are arranged in any random topology. For data lookup, flooding is necessary. In unstructured systems, problems brought on by frequent peer joining and leaving the system, or "churn", are effectively handled. However, this compromises the effectiveness of data querying and the crucial flexibility. Lookups are not guaranteed in unstructured networks. On the other hand, structured overlay networks offer deterministic limits on data discovery. They create scalable network overlays based on a distributed data structure that truly allows deterministic data lookup behavior.

The usage of distributed hash tables (DHTs) is a recent trend in the design of structured overlay systems [5, 9, 19]. Overlay designs of this type can provide efficient, flexible, and resilient service [5, 9, 11, 19, 20]. However, addressing the churn issue while maintaining DHTs becomes expensive. It requires significant transformation in designing an effective data query service. There are numerous notable publications in this area that have examined creating hybrid systems [4, 14, 17, 21]. These works make an attempt to incorporate the benefits of both structured and unstructured structures. These works, however, have their own set of advantages and disadvantages [1].

The study and application of encrypted communication protocols is known as cryptography. It is concerned with the development and examination of protocols that prohibit harmful third parties from gaining access to information transferred between two companies, hence complying to many principles of information security. Secure communication refers to a scenario in which a message or data transmitted between two parties cannot be accessed by an adversary. An adversary in cryptography is a malevolent actor who seeks to get usable information or data by violating information security rules. Cryptographic techniques are used to ensure authentication and confidentiality stability in peer-to-peer networks. The two most common forms of cryptographic algorithms are secret key cryptographic algorithms and public key cryptographic algorithms. Secret key cryptographic algorithms are also known as symmetric key algorithms since the same key is used for encryption and decryption and is shared by all parties involved. Public key cryptography algorithms, on the other hand, are also known as asymmetric key algorithms. This version employs a pair of keys, one for encryption and the other for decryption [12].

In [8] the authors have used a well-defined approach of combining the symmetric key and the public key cryptography methods to ensure security in the architecture. The main issue with symmetric key encryption is secured key delivery among sender and receiver. This requires reliable communication network that are completely resistant to any form of privacy attacks which can be expensive and often highly unlikely in the context of P2P network. Shared symmetric key can be vulnerable to any man-in-the-middle attack as the same key is being used reused for encryption and decryption. In[16] the authors suggested robust public-key cryptography approaches for the security of current communication protocols in [8] with

comparably lower costs to overcome the aforementioned issues with employing symmetric key. They have added anonymity to these strategies as well.

However, asymmetric encryption can be more expensive than symmetric encryption but it does allow us to overcome the downfall's of using symmetric encryption. Similarly the authors in [16] have considered both an intra-group and an inter-group lookup technique in an residue class(RC)-based architecture [8] with secured protocols. Asymmetric key cryptography has been used with public keys. Moreover, they have suggested secured capacity-constrained multicast algorithms for the two-level architecture as well as for use inside groups. They have also considered anonymity. We have defined the RC based architecture in section 2.

The authors of this [16] article presupposed that all of the peers can be trusted. This situation has been taken into consideration in this work. The main communication node in the RC-based system is the group-heads. The invader will try to breach the group-heads in order to destabilize the entire network. In this work, we provide secure data-lookup algorithms while considering the aforementioned threat models.

1.1 Our Contribution

In this research, interest-based P2P systems have been taken into consideration [21, 22]. We have thought about creating secure protocols for both intra and inter group lookup algorithm. Public keys for asymmetric key cryptography and digital signature authentication have been utilized. The rest of the paper is organized as follows. We give the preliminaries and the overview of the RC-based 2-level non-DHT-based structured P2P network proposed in [8] in section 2. The secured data lookup and transfer algorithms for both the inter and the intra group along with the threat model are explained in section 3. In section 4, we present the performance evaluation. Finally, we conclude in section 5.

2 Preliminaries

Here, we have taken into consideration some of the first results of an RC-based low diameter two level hierarchical structured P2P network [6, 7, 13]. We provide a structured design for an interest-based peer-to-peer system in this section. We will use the following notations and their meanings to 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 $S = \{P^{Ri}\}$, $0 \leq i \leq n-1$, where P^{Ri} denotes the subset consisting of all peers with the same resource type Res_i. and the number of distinct resource types present in the system is n. Also, for each subset P^{Ri}, we assume that G_i^h is the first peer among the peers in P^{Ri} to join the system. We call G_i^h 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 [7].

2.1 Residue Class

Modular arithmetic has been used to define the residue class(RC)-based architecture of the P2P system. Consider the set S_n of non negative 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, $a \equiv r(modn)\}$.

For example, for n = 3, the classes are:
[0] = {..., - 6, - 3, 0, 3, 6, ...}
[1] = {..., - 5, - 2, 1, 4, 7, ...}
[2] = {..., - 4, - 1, 2, 5, 8, ...}

Each integer used to represent a residue class in the P2P architecture serves as the logical (overlay) address for the group-head of a group. For example, logical address of the first group-head is 0, for the second one it is 1, and so on. The logical (overlay) addresses of peers with a common interest (i.e., peers in the same group) are represented by integers belonging to various classes, and the number of residue classes corresponds to the number of different resource types. For the sake of simplicity, only positive integer values are used for addressing. It becomes obvious that any class may technically contain an endless number of integers, which indicates that there is no upper limit on the size of a group.

2.2 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 G_i^h $(0 \leq i \leq n-1)$. 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}, $(0 \leq i \leq n-1)$, such that all peers $(\in P^{Ri})$ 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 G_i^h.
3. Based upon whether a peer in the network, a group-head or a regular peer, they maintain Information Resource Table (IRT) and Local Resource Table (LRT) that consists of n number of tuples. The group-heads maintains both Information Resource Table (IRT) and Local Resource

Table (LRT), where IRT contains list of the other group-heads, while LRT contains list of the peers present in a group. The regular peers in a group only maintain the LRT.

- The group heads will have a tuple of the form <Group Head Logical Address, Group Head IP Address, Group Head public key>for other group heads and <Peer Logical Address, Peer IP Address, peer public key>for the other peers present in the network. The Group Head Logical Address are assigned according to the proposed logical address assignment algorithm proposed in section 2.2 and the public key of the group heads or the peers are exchanged when they are joining the network and the IRT is updated and broadcasted in the network. Also, Resource Code is the same as the group head logical address.
- The peers P_i, who are not group heads but belongs to a group G_i ($P_i \in G_i$) will have the tuple of the form <Group Head Logical Address, Group Head public Key>for group head of G_i and <Peer Logical Address, Peer public Key>for the other peers present in G_i.

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 G_x^h and G_y^h.

The proposed architecture is shown in Figure 1.

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 = $\{P^{Ri}\}$,$(0 \leq i \leq n-1)$. Also, as mentioned earlier, for each subset P^{Ri} (i.e. group G_i) peer G_i^h 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. IRT remains sorted with respect to the logical addresses of the group-heads.

Definition 3. Two peers G_i^h and G_j^h on the ring network are logically linked together if (i + 1) mod 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 1. Diameter of the transit ring network is $n/2$.

Lemma 2. 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.
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. Every resource type will have two group-heads, to maintain a fault-tolerance system and secure communications which are explained in the following sections.
6. 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.

3 Data Lookup Algorithms with Public Key Cryptography

This part introduces data lookup techniques [6, 13], both intra and inter, with the idea of security using public key cryptography and digital signature for authentication. Figure 2 explains how cryptographic operations are used in a two-level RC-based design.

3.1 Public key distribution among group-heads and peers

We begin with a straightforward technique that allows all group heads to be aware of one another's public keys. Suppose that there are now k groups in the network, with the group leaders' logical addresses ranging from 0 to (k-1). Specifically, the biggest resource code available right now in IRT is (k-1). The group head G_0^h initially determines if the joining peer's resource type is already available in the IRT when a peer p having instance(s) of a certain resource type requests to join. Now, one of the two subsequent circumstances may happen.

Situation 1: Resource type of peer p does not exist in IRT

1. New peer p contacts G_0^h for joining the network.
2. IRT holds resource codes from 0 up to (k-1) before peer p joins, therefore the group-head G_0^h assigns the joining peer with the next greatest available number for the code; as a result, the code for p becomes k.
3. G_0^h makes entry in the IRT for the new resource code k (which is now the logical address of the newly joining peer p) and the IP address of peer p, because now peer p becomes the group-head G_k^h.
4. G_0^h and G_k^h exchange their public keys, namely PU_0 and PU_k. Their respective private keys are PR_0 and PR_k. Group-head G_0^h updates a list T (IRT) by including PU_k

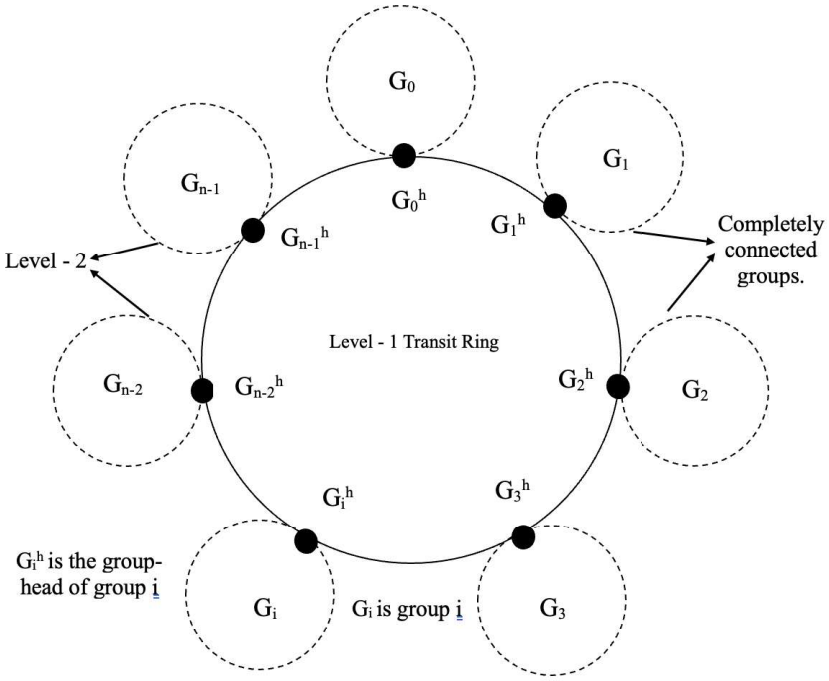

Figure 1: A two-level RC based structured P2P architecture with n distinct resource types

in it. List T now contains (k+1) different public keys corresponding to (k +1) group-heads. List T initially contains only PU_0 of G_0^h.

5. G_0^h performs E (T, PR_0k) and executes the broadcast protocol so that each node (group-head) on the transit-ring receives a copy of the encrypted IRT T; each receiving node performs D (PU_0, E (T, PR_0k)) to get the recent copy of the IRT T. Now each node has the knowledge of the public keys of all existing nodes.
6. The above five steps are executed each time a peer with a new resource type contacts G_0^h to join the network.

Situation 2: Resource type of peer p exists in IRT

The following procedures are carried out in order to add the new peer p to the network when it tries to join the system and has some instance(s) of an existing resource type with code, say, m.

1. New peer p contacts G_0^h for joining the network
2. G_0^h checks with the (m+1)th entry in IRT to get the IP address of G_m^h
3. G_0^h sends the IP address of G_m^h to p
4. Peer p contacts G_m^h
5. If peer p is the second peer to join a group of a particular resource code m, then G_m^h will assign peer p as a secondary group-head for resource code m.
   (a) G_m^h gives its public key PU_m to peer p and peer p gives it's public key pub_i to G_m^h
   (b) G_m^h will update its IRT with peer p's information < Peer p's logical address, Peer p's IP, Peer p's public-key GS_m^h >and will broadcast it to all the other group-heads.
   (c) G_m^h update its local resource table (LRT) and broadcast it in the group
6. If peer p is not the second peer to join a group of a particular resource code m
   (a) G_m^h gives its public key PU_m to peer p and peer p gives it's public key pub_i to G_m^h
   (b) G_m^h update its local resource table (LRT) and broadcast it in the group

3.2 Intra Group Lookup Algorithm in RC Based Architecture with Public Key Security

The resource lookup occurs within the group in this scenario, which means that the resource type is the same for both parties but the value is different. The algorithm for intra group data lookup is explained as follows.

Let us assume that in a group G_y, a peer $G_{y,i}$ with the resource $<Res_y, V_i>$is querying for a resource $<Res_y, V_j > Req_j$. The requesting peer will broadcast the request message Req_j in the group G_i using the LRT table as explained in Algorithm 1.

Observation 1: There will be two group-heads (G_i^h and GS_i^h) if there are more than two peers in a group (G_i) creating information redundancy for attack mitigation. All of the other group-heads and the group members have knowledge to this

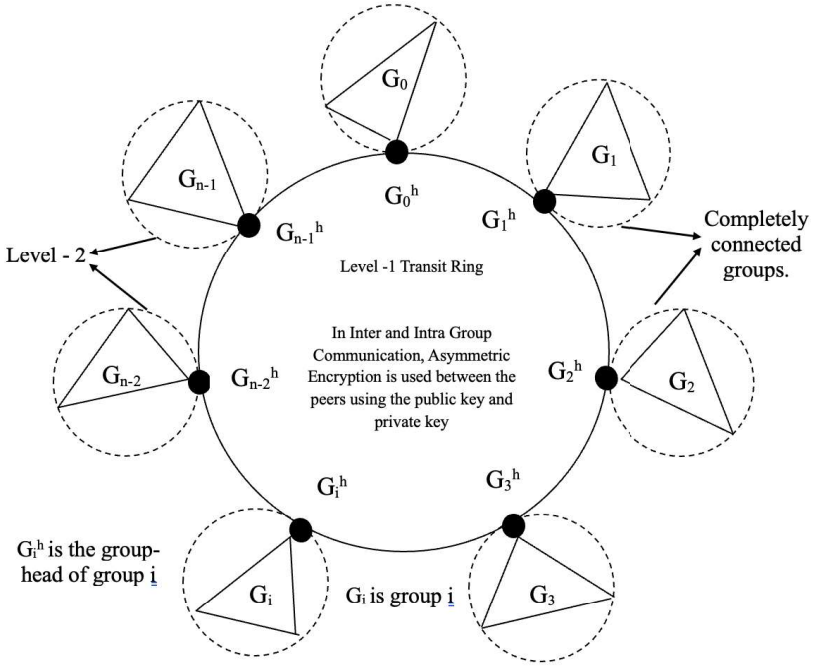

Figure 2: Inter and intra group communication using asymmetric encryption

Algorithm 1 Intra Group Lookup Algorithm in RC Based Architecture with Public Key Security

1: $G_{y,i}$ broadcast the request message Req_j in G_i using the IRT table
2: **if** $\ni G_{y,j} \in G_y$ with Req_j **then**
3: $G_{y,j}$ encrypts Res_j: E(Res_j, $PU_{G_{y,i}}$) ▷ $G_{y,j}$ gets the $PU_{G_{y,i}}$ from IRT
4: Unicast the encrypted message E(Res_j, $PU_{G_{y,i}}$)
5: $G_{y,i}$ decrypts the message: D(E, PR_i)
6: **else**
7: Search for $<Res_y, V_j>$fails
8: **end if**

information.

Observation 2: The maximum number of hops required to locate a resource using the Intra Group Lookup Algorithm in RC Based Architecture with Public Key Security is 2.

3.3 Inter Group Lookup Algorithm in RC Based Architecture with Public Key Security

Inter group communication takes place between nodes from two separate interest-based groups. Any communication between the peers $G_{x,i} \in$ group G_x and $G_{y,j} \in$ group G_y in our public-key based secure RC-based architecture occurs solely through the appropriate group heads G_x^h and G_y^h.

Let us assume that a peer $G_{x,i} \in$ group G_x with the resource $<Res_x, V_i>$ is querying for a resource $<Res_y, V_j>$.Both the peer $G_{x,i}$ and the group head G_x^h are aware about the fact that $Res_y \notin G_x$.

In this case 2 scenarios can happen:

1. The group-head $G_{y,j} \in$ group G_y has $<Res_y, V_j>$.
2. A peer $P_i \in$ group G_y has $<Res_y, V_j>$.

Case 1: The group-head $G_{y,j} \in$ group G_y has $<Res_y, V_j>$. The algorithm for the secured public-key based data lookup search for case 1 in RC Based Architecture is explained in Algorithm 2.

Observation 3: The maximum number of hops required to locate a resource using the Inter Group Lookup Algorithm in RC Based Architecture with Public Key Security for case 1 is n/2 + 7, where n is the total number of resource present in the network.

Case 2: The group-head $G_{y,j} \in$ group G_y has $<Res_y, V_j>$. The algorithm for the secured public-key based data lookup search for case 1 in RC Based Architecture is explained in Algorithm 3.

Observation 4: The maximum number of hops required to locate a resource using the Inter Group Lookup Algorithm in RC Based Architecture with Public Key Security for case 2 is n/2 + 11, where n is the total number of resource present in the network..

3.4 Threat model

We begin by describing the nature of attack on the peer-to-peer system along with the intruder objective and capabilities. This threat model emphasizes the vulnerability of the group heads and disruptions in inter-group communication due to insider

Algorithm 2 : Case 1: $G_{y,j} \in$ group G_y has $<Res_y, V_j>$

1: $G_{x,i}$ encrypts the request message E($<Res_y, V_j>$, $PU_{G_x^h}$) using the LRT table
2: G_x^h decrypts the request message D(E, $PR_{G_x^h}$)
3: G_x^h looks at its IRT and find $G_y^h : Res_y \in G_y^h$
4: G_x^h encrypts the request message E(Res_y, $PU_{G_y^h}$) using the IRT table and broadcast it in the transit ring
5: G_y^h decrypts the request message D(E, $PR_{G_y^h}$) ▷ $G_y^h \in$ group G_y holds the resource type of Res_y
6: **if** G_y^h itself have $<Res_y, V_j>$ **then**
7: G_y^h performs hash of its $PU_{G_y^h}$: H($PU_{G_y^h}$)
8: G_y^h unicast E($PU_{G_y^h}$ + H($PU_{G_y^h}$), $PU_{G_x^h}$) to G_x^h
9: G_x^h decrypts the message D(E, $PR_{G_x^h}$)
10: G_x^h encrypts with message E($PU_{G_{x,i}}$, $PU_{G_y^h}$ + H($PU_{G_y^h}$) and unicast it to $G_{x,i}$
11: $G_{x,i}$ decrypts the message with D(E, $PR_{G_{x,i}}$)
12: $G_{x,i}$ performs hash function on received public-key: H($PU_{G_y^h}$)
13: $G_{x,i}$ checks H($PU_{G_y^h}$) with the received hash value in step 11
14: **if** step 13 == true **then**
15: $G_{x,i}$ encrypts $PU_{G_{x,i}}$ with $PU_{G_y^h}$ and unicast it to G_x^h: E($PU_{G_{x,i}}$, $PU_{G_y^h}$)
16: G_x^h unicast it to G_y^h
17: G_y^h performs D(E, $PR_{G_y^h}$)
18: G_y^h perform E($<Res_y, V_j>$, $PU_{G_{x,i}}$) and unicast it to G_x^h
19: G_x^h forward the message to $G_{x,i}$
20: $G_{x,i}$ performs D(E, $PR_{G_{x,i}}$)
21: **else**
22: Algorithm 4
23: **end if**
24: **else**
25: Case 2
26: **end if**

or backdoor access attack. Due to the one-hop transfer to the members, the intra-group communication remains unaffected while the intruder takes covert control of the *inter-group communication in the transit ring network*.

The objective of the intruder is to cause Loss of Integrity by *compromising selective group heads that corrupts the message data*. This is an extremely realistic high-impact attack as the adversary acquires and maintains covert access on the query-and-reply process with remote administration tool or rootkit. We assume that the intruder has no control over the members. We can experiment/explore the attack impact by varying the number of compromised group heads and extent of data modifications. We want to highlight network operations and components that remain unaffected during the attack. It is assumed that the intruder does not make large data modifications in the Information Resource Table, because it can be easily verified among the neighboring group heads with fast look-ups.

In addition, the encryption at the group head and member levels should operate correctly which follows mathematically proven robust behavior of asymmetric encryption. As public key (redundancy) is maintained in all group heads and hence, such large-scale covert data modification is unlikely. In section 3.5 for both the cases mentioned in section 3.3 we present a solution to the above described difficulty of a malicious group-head by adding the notion of assigning a second group-head to each group.

If there are more than two peers in a group (G_i), there will be two group-heads (G_i^h and GS_i^h), generating information redundancy for attack mitigation. The other group-heads and members are all aware of this information. If step 13 of algorithm 2 or algorithm 3 returns false, it indicates that one or more malicious nodes (the group-heads) are present in the path between the destination node and the source node, because the hash value received from the destination node differs from the hash value obtained by the source after performing the hash on the destination node's public key. It's also possible that the destination node is malicious as well.

In this circumstance, we propose the algorithms for carrying on the communication with the second group-heads in section 3.5. We terminate communication if the aforementioned hash values are still out of sync. At this point, it is possible to recommend adding a third group head, a fourth group head, and so on. But, doing so would result in an increase in the size of the IRT table, which would slow down memory access as well

Algorithm 3 : Case 2: A peer $P_i \in$ group G_y has $<Res_y, V_j>$

$G_{x,i}$ encrypts the request message E($<Res_y, V_j>$, $PU_{G_x^h}$) using the LRT table
G_x^h decrypts the request message D(E, $PR_{G_x^h}$)
G_x^h looks at its IRT and find $G_y^h : Res_y \in G_y^h$
G_x^h encrypts the request message E(Res_y, $PU_{G_y^h}$) using the IRT table and broadcast it in the transit ring
G_y^h decrypts the request message D(E, $PR_{G_y^h}$) ▷ $G_y^h \in$ group G_y holds the resource type of Res_y
if G_y^h does not have $<Res_y, V_j>$ **then**
    G_y^h broadcast $<Res_y, V_j>$in group G_y
    if $P_i \in$ group G_y has $<Res_y, V_j>$ **then**
        P_i performs hash of its PU_{P_i}: H(PU_{P_y})
        P_i unicast E(PU_{P_i} + H(PU_{P_i}), $PU_{G_y^h}$) to G_y^h
        G_y^h decrypts the message D(E, $PR_{G_y^h}$)
        G_y^h encrypts: E(PU_{P_i} + H(PU_{P_i}), $PU_{G_x^h}$)
        G_x^h decrypts the message D(E, $PR_{G_x^h}$)
        G_x^h encrypts with message E($PU_{G_{x,i}}$, $PU_{G_y^h}$ + H($PU_{G_y^h}$) and unicast it to $G_{x,i}$
        $G_{x,i}$ decrypts the message with D(E, $PR_{G_{x,i}}$)
        $G_{x,i}$ performs hash function on received public-key: H($PU_{G_y^h}$)
        $G_{x,i}$ checks H($PU_{G_y^h}$) with the received hash value in step 11
        if step 13 == true **then**
            $G_{x,i}$ encrypts $PU_{G_{x,i}}$ with $PU_{G_{y,i}}$ and unicast it to G_x^h: E($PU_{G_{x,i}}$, $PU_{G_{y,i}}$
            G_x^h unicast it to G_y^h
            G_y^h uincast to $G_{y,i}$
            $G_{y,i}$ performs D(E, $PR_{G_{y,i}}$)
            $G_{y,i}$ perform E($<Res_y, V_j>$, $PU_{G_{x,i}}$) and unicast it to G_y^h
            G_y^h unicast message to G_x^h
            G_x^h forward the message to $G_{x,i}$
            $G_{x,i}$ performs D(E, $PR_{G_{x,i}}$)
        else
            Algorithm 5
        end if
    end if
else
    Search for $<Res_y, V_j>$has failed
end if

as data-lookup episodes. The number of stand-by group leaders needed will therefore largely depend on the network designer, who will make this decision after considering the scenario.

4 Performance Evaluation

P2P networks offer the potential to improve network capabilities through the sharing of music, video, and other services. Nonetheless, P2P networks pose security issues since the nodes are exposed to a variety of security attacks. One of them is the man-in-the-middle attack. It is an indirect intrusion attempt in which the attacker places his computing device between two nodes. As a result, the intermediary node can intercept and change communications sent between two valid users without the knowledge of the sender and recipient. Encryption technologies can be used for data transmission as a defense measure.

The algorithms we proposed for data-lookup in section 3 are immune to such attacks because every message communicated is encrypted with the public key of the immediate destination peer, and the message can be decrypted by only the destination peer with its own private key. This prohibits anybody along the path of the message from interfering with the communication between the source and the destination.

To understand trade-off analysis between efficiency vs effectiveness of security measures, we must estimate the tri-factor performance cost related to computation, memory usage and network utilization. Cryptographic operations for signature generation and verification is inexpensive as key size is much smaller than the actual data. Our protocol generates the

Algorithm 4 : Case 1: $G_{y,j} \in$ group G_y has $<Res_y, V_j>$

1: $G_{x,i}$ encrypts the request message E($<Res_y, V_j>$, $PU_{GS_x^h}$) using the LRT table
2: GS_x^h decrypts the request message D(E, $PR_{GS_x^h}$)
3: GS_x^h looks at its IRT and find $GS_y^h : Res_y \in GS_y^h$
4: GS_x^h encrypts the request message E(Res_y, $PU_{GS_y^h}$) using the IRT table and broadcast it in the transit ring
5: GS_y^h decrypts the request message D(E, $PR_{GS_y^h}$) ▷ $GS_y^h \in$ group G_y holds the resource type of Res_y
6: **if** GS_y^h itself have $<Res_y, V_j>$ **then**
7: GS_y^h performs hash of its $PU_{GS_y^h}$: H($PU_{GS_y^h}$)
8: GS_y^h unicast E($PU_{GS_y^h}$ + H($PU_{GS_y^h}$), $PU_{GS_x^h}$) to GS_x^h
9: GS_x^h decrypts the message D(E, $PR_{GS_x^h}$)
10: GS_x^h encrypts with message E($PU_{G_{x,i}}$, $PU_{GS_y^h}$ + H($PU_{GS_y^h}$) and unicast it to $G_{x,i}$
11: $G_{x,i}$ decrypts the message with D(E, $PR_{G_{x,i}}$)
12: $G_{x,i}$ performs hash function on received public-key: H($PU_{GS_y^h}$)
13: $G_{x,i}$ checks H($PU_{GS_y^h}$) with the received hash value in step 11
14: **if** step 13 == true **then**
15: $G_{x,i}$ encrypts $PU_{G_{x,i}}$ with $PU_{GS_y^h}$ and unicast it to GS_x^h: E($PU_{G_{x,i}}$, $PU_{GS_y^h}$)
16: GS_x^h unicast it to GS_y^h
17: GS_y^h performs D(E, $PR_{GS_y^h}$)
18: GS_y^h perform E($<Res_y, V_j>$, $PU_{G_{x,i}}$) and unicast it to GS_x^h
19: GS_x^h forward the message to $G_{x,i}$
20: $G_{x,i}$ performs D(E, $PR_{G_{x,i}}$)
21: **else**
22: Algorithm 3
23: **end if**
24: **else**
25: Case 2
26: **end if**

signatures only once when we are transferring the public key. Similarly, verification also happens at the member level which democratizes the detection and further defense tasks.

A more robust solution of multi-member verification within the intra-group may incur additional processing costs leading to unnecessary detection latency. Fast comparison between calculated vs received signature at each level reduces detection delays and allows for easier mitigation and recovery. Such corrective functionality needs to be implemented at the member-level as the extent of corruption at the group heads remains unclear during the attack. Detailed exploration of the mitigation framework demands in-depth exploration and is skipped for brevity in this work.

In our signature-based authentication at the member-level, the signature values are immediately shared with the member nodes eliminating the storage needs. There is no overhead of control messages for maintaining network security since when a peer joins the network, they update the IRT with their ip_address, resource code, and public key information, which is then broadcast across the network. As a result, following the broadcast, everyone in the network will have access to the most up-to-date information. By the same reasoning, bandwidth utilization remains unaffected as no additional message bits are exchanged during the regular broadcasting

A potential vulnerability in the proposed network is the disruption of the group head, which can result in a single point of failure during a man-in-the-middle attack. The group head plays a crucial role in coordinating and facilitating communication within a specific group or subgroup of peers. If the group head is disrupted or compromised, it can severely impact inter-group communication and hinder the overall functionality of the network. As an alternative solution, redundancy measures such as multiple group heads need incur additional hardware and communication cost.

As soon as a peer enters the network, they update the IRT with their IP address, resource code, and public key information, which is then broadcast throughout the network, there is no need for control messages to ensure network security. As a result, everyone on the network will have access to the most recent information after the broadcast. The data-lookup complexity of the proposed public-key and digital signature based secured data-lookup protocols is O(n), n $<<$N, where n is the total number of resources present in the network and N is the total

Algorithm 5 : Case 2: A peer $P_i \in$ group G_y has $<Res_y, V_j>$

$G_{x,i}$ encrypts the request message E($<Res_y, V_j>$, $PU_{GS_x^h}$) using the LRT table
GS_x^h decrypts the request message D(E, $PR_{GS_x^h}$)
GS_x^h looks at its IRT and find $GS_y^h : Res_y \in GS_y^h$
GS_x^h encrypts the request message E(Res_y, $PU_{GS_y^h}$) using the IRT table and broadcast it in the transit ring
GS_y^h decrypts the request message D(E, $PR_{GS_y^h}$) ▷ $GS_y^h \in$ group G_y holds the resource type of Res_y
if GS_y^h does not have $<Res_y, V_j>$ **then**
    GS_y^h broadcast $<Res_y, V_j>$in group G_y
    if $P_i \in$ group G_y has $<Res_y, V_j>$ **then**
        P_i performs hash of its PU_{P_i}: H(PU_{P_i})
        P_i unicast E(PU_{P_i} + H(PU_{P_i}), $PU_{GS_y^h}$) to GS_y^h
        GS_y^h decrypts the message D(E, $PR_{GS_y^h}$)
        GS_y^h encrypts: E(PU_{P_i} + H(PU_{P_i}), $PU_{GS_x^h}$)
        GS_x^h decrypts the message D(E, $PR_{GS_x^h}$)
        GS_x^h encrypts with message E($PU_{G_{x,i}}$, $PU_{GS_y^h}$ + H($PU_{GS_y^h}$) and unicast it to $G_{x,i}$
        $G_{x,i}$ decrypts the message with D(E, $PR_{G_{x,i}}$)
        $G_{x,i}$ performs hash function on received public-key: H($PU_{GS_y^h}$)
        $G_{x,i}$ checks H($PU_{GS_y^h}$) with the received hash value in step 11
        if step 13 == true **then**
            $G_{x,i}$ encrypts $PU_{G_{x,i}}$ with $PU_{G_{y,i}}$ and unicast it to GS_x^h: E($PU_{G_{x,i}}$, $PU_{G_{y,i}}$
            GS_x^h unicast it to GS_y^h
            GS_y^h uincast to $G_{y,i}$
            $G_{y,i}$ performs D(E, $PR_{G_{y,i}}$)
            $G_{y,i}$ perform E($<Res_y, V_j>$, $PU_{G_{x,i}}$) and unicast it to GS_y^h
            GS_y^h unicast message to GS_x^h
            GS_x^h forward the message to $G_{x,i}$
            $G_{x,i}$ performs D(E, $PR_{G_{x,i}}$)
        else
            Algorithm 4
        end if
    end if
else
    Search for $<Res_y, V_j>$has failed
end if

number of peers present in the network. The complexity of the data lookup is not dependent on the total number of peers in the network. This improves the robustness and scalability of our network.

5 Conclusion

In this work, a 2-level non-DHT-based P2P architecture was considered. The choice to utilize an interest-based architecture was made because:

1. We have previously shown [7] that the data lookup strategies outperform many really well-known DHT-based contributions [15, 18, 23] in terms of search latency.
2. It is beneficial over several existing interest-based systems [1, 3, 5, 9, 10, 19]. Public-key cryptography and hashing has been used to successfully integrate security into the intra-group and inter-group communication techniques described in [7] in this work.

To guarantee the security of the architecture, the writers of [8] combined the symmetric key and public key cryptography approaches using a well-defined methodology. Secured key delivery between sender and receiver is the primary drawback of symmetric key encryption. Using a single shared key exposes the network vulnerability on both sides due to attack leading to loss of integrity . In [16] the authors used asymmetric key cryptography to address the aforementioned problems and offered strong public-key cryptography methodologies for the security of contemporary communication protocols in [8] with correspondingly reduced costs.

In this paper, we have presented a public key cryptography

and digital signature authentication based data lookup protocol that are efficient and secured in RC-based hierarchical structured P2P network. Adversary takes covert control of the inter-group communication in the transit ring network as part of the threat model. Our scheme compares digital signatures generated with assymetric key for authenticating the group heads. We are assuming that the peers are trusted and functioning correctly as in the earlier article [16]. The proposed public-key and digital signature based secured data-lookup protocols have a data-lookup complexity of $O(n)$, $n << N$, where n is the total number of resources in the network and N is the total number of peers in the network. The overall number of peers in the network has no impact on how costly the data lookup is. As a result, our network is more reliable and scalable. Using the algorithms presented in this research, we can detect presence of malicious nodes in the network but we cannot identify them. As a follow-up research, we are thinking of creating a trust algorithm to be used amongst group-heads in order to rule out malicious group-heads.

References

[1] Lyes Badis, Mourad Amad, Djamil Aïssani, Kahina Bedjguelal, and Aldja Benkerrou. "ROUTIL: P2P Routing Protocol Based on Interest Links". In *2016 International Conference on Advanced Aspects of Software Engineering (ICAASE), IEEE, pp. 1 - 5*, 2016.

[2] Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau, Nick Lanham, and Scott Shenker. "Making Gnutella-like P2P Systems Scalable". In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, New York, NY, USA. Association for Computing Machinery, pp. 407-418, 2003.

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

[4] Prasanna Ganesan, Qixiang Sun, and Hector Garcia-Molina. "Yappers: A Peer-to-Peer Lookup Service Over Arbitrary Topology". In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, 2: 1250–1260, 2003.

[5] Mo Hai and Yan Tu. "A P2P E-Commerce Model Based on Interest Community". In *2010 International Conference on Management of e-Commerce and e-Government, IEEE, pp. 362-365*, 2010.

[6] Swathi Kaluvakuri, Bidyut Gupta, Banafsheh Rekabdar, Koushik Maddali, and Narayan Debnath. "Design of RC-Based Low Diameter Two-Level Hierarchical Structured P2P Network Architecture". In Mohammed Serrhini, Carla Silva, and Sultan Aljahdali, editors, *Innovation in Information Systems and Technologies to Support Learning Research, pp. 312-320*, Springer International Publishing, pp. 312-320, 2020.

[7] 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 Computer and Their Applications pp. 74*, 2020.

[8] Swathi Kaluvakuri, Indranil Roy, Koushik Maddali, Bidyut Gupta, and Narayan Debnath. "Efficient Secured Data Lookup and Multicast Protocols with Anonymity in RC-Based Two-level Hierarchical Structured P2P Network.". *International Journal for Computers & Their Applications*, 28(3):140-149, 2021.

[9] Mujtaba Khambatti, Kyung Dong Ryu, and Partha Dasgupta. "Structuring Peer-to-Peer Networks Using Interest-based Communities". In *International Workshop On Databases, Information Systems, and Peer-to-Peer Computing, Springer, pp. 48–63*, 2003.

[10] Sardar Kashif Ashraf Khan and Laurissa N Tokarchuk. "Interest-based Self Organization in Group-Structured P2P Networks". In *2009 6th IEEE Consumer Communications and Networking Conference, pp. 1-5*, 2009.

[11] Dmitry Korzun and Andrei Gurtov. "Hierarchical Architectures in Structured Peer-to-Peer Overlay Networks". *Peer-to-Peer Networking and Applications, Springer*, 7(4):359-395, 2014.

[12] Koushik Maddali, Swathi Kaluvakuri, Nick Rahimi, Bidyut Gupta, and Narayan Debnath. "On Designing Secured Communication Protocols along with Anonymity for CRT based Structured P2P Network Architecture". *EPiC Series in Computing*, 75:59-68, 2021.

[13] Koushik Maddali, Banafsheh Rekabdar, Swathi Kaluvakuri, and Bidyut Gupta. "Efficient Capacity-Constrained Multicast in RC-Based P2P Networks". In *Proceedings of 32nd International Conference on*, 63:121-129, 2019.

[14] Zhuo Peng, Zhenhua Duan, Jian-Jun Qi, Yang Cao, and Ertao Lv. "HP2P: A Hybrid Hierarchical P2P Network". In *First International Conference on the Digital Society (ICDS'07), IEEE, pp. 18*, 2007.

[15] Antony Rowstron and Peter Druschel. "Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-Peer Systems". *Distributed Systems Platforms (Middleware),In IFIP/ACM International Conference, pp. 329-350*, 2001.

[16] Indranil Roy, Nick Rahimi, Reshmi Mitra, and Swathi Kaluvakuri. "Efficient Secured Public-Key based Data Lookup and Multicast Protocols with Anonymity in RC-Based Two-level Hierarchical Structured P2P Network". In *Proceedings of 35th International Conference on Computer Applications in Industry and Engineering*, EPiC Series in Computing, EasyChair, 89:62-71, 2022.

[17] Kai Shuang, Peng Zhang, and Sen Su. "Comb: A Resilient and Efficient Two-hop Lookup Service for Distributed Communication System". *Security and Communication Networks*, 89:62-71, 2022.

[18] 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 Transactions on networking, pp. 17–32*, 11(1):17-31, 2003.

[19] Zhiyong Tu, Wei Jiang, and Jinyuan Jia. "Hierarchical Hybrid DVE-P2P Networking Based on Interests Clustering". In *2017 International Conference on Virtual Reality and Visualization (ICVRV), IEEE, pp. 378–381*, 2017.

[20] Ming Xu, Shuigeng Zhou, and Jihong Guan. "A New and Effective Hierarchical Overlay Structure for Peer-to-Peer Networks". *Computer Communications, Elsevier*, 34(7):862-874, 2011.

[21] Min Yang and Yuanyuan Yang. "An Efficient Hybrid Peer-to-Peer System for Distributed Data Sharing". *IEEE Transactions on computers, IEEE*, 59(9):1158-1171, 2009.

[22] Rongmei Zhang and Y Charlie Hu. "Assisted Peer-to-Peer Search With Partial Indexing". *IEEE Transactions on Parallel and Distributed Systems, IEEE*, 18(8):1146-1158, 2007.

[23] Ben Y Zhao, Ling Huang, Jeremy Stribling, Sean C Rhea, Anthony D Joseph, and John D Kubiatowicz. "Tapestry: A resilient global-scale overlay for service deployment". *IEEE Journal on selected areas in communications, IEEE*, 22(1):41-53. 2004.

Anjila Neupane (photo not available) is pursuing her masters in Applied Computer Science from Southeast Missouri State University. She have completed her Bachelors Degree in Computer Engineering from Tribhuwan University in Kathmandu, Nepal. Her primary research interests revolve around peer-to-peer (P2P) networks and their applications. She is particularly interested in exploring the potential of P2P systems to improve communication and information sharing among individuals and organizations.

Indranil Roy (photo not available) is an Assistant Professor in the Department of Computer Science at the Southeast Missouri State University. He received his MS and Ph.D. degrees in Computer Science from Southern Illinois University, Carbondale in 2018 and 2022, respectively. His current research interest includes the design of architecture and communication protocols for structured peer-to-peer overlay networks, security in overlay networks, and Blockchain.

Reshmi Mitra (photo not available) is an Assistant Professor in the Department of Computer Science at the Southeast Missouri State University. She received her MS and Ph.D. degrees in Electrical and Computer Engineering from the University of North Carolina at Charlotte in 2007 and 2015, respectively. Previously she has worked at the National Institute of Technology India, Advanced Micro Devices Austin, and Samsung Austin R&D Center. Her research interests include Security and Performance issues in IoT, Cloud Computing, and Blockchain.

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.