SECURITY CONSIDERATION IN PEER-TO-PEER NETWORKS WITH A CASE STUDY APPLICATION

Nick Rahimi

School of Information Systems & Applied Technology Southern Illinois University, Carbondale, IL

ABSTRACT

Peer-to-Peer (P2P) overlay networks wide adoption has also created vast dangers due to the millions of users who are not conversant with the potential security risks. Lack of centralized control creates great risks to the P2P systems. This is mainly due to the inability to implement proper authentication approaches for threat management. The best possible solutions, however, include encryption, utilization of administration, implementing cryptographic protocols, avoiding personal file sharing, and unauthorized downloads. Recently a new non-DHT based structured P2P system is very suitable for designing secured communication protocols. This approach is based on Linear Diophantine Equation (LDE) [1]. The P2P architectures based on this protocol offer simplified methods to integrate symmetric and asymmetric cryptographies' solutions into the P2P architecture with no need of utilizing Transport Layer Security (TLS), and its predecessor, Secure Sockets Layer (SSL) protocols.

KEYWORDS

Decentralized Network, Peer-to-Peer, Security, Encryption, Distributed Hash Tables & Linear Diophantine

1. INTRODUCTION

The Internet has evolved a lot in the past decades. Emerging the Internet of Things (IoT), takes the data communication to a completely new level. Today, a traditional network such as client-server or content-delivery based networks (CDN) are not capable to provide the daily or even hourly necessities of the end-users around the world. Additional technologies such as P2P are essential for solving these limitations. P2P overlay network that constructs on top of a traditional underlying network, enables developers to create and implement protocols such as data routing or file sharing on the web easily. The cooperative model of P2P architecture and the fact that their users (peers) bring their own resource to share, provides a set of remarkable benefits. These types of overlay systems are highly scalable due to the idea that the capacity of their resources such as storage, processors, and bandwidth escalates proportionally with the number of users. The load of the network spreads across the peers; consequently, the probability of all nodes being crashed at the same time would be unprecedented if not impossible. Furthermore, the distribution of the peers delivers better efficiency. A resource can be located in a nearby peer and this advantage can save a lot of bandwidth and time consumption. Additionally, all the peers in a P2P system are powered with equal abilities, accountabilities, and functionalities, despite their different possessions. However, most of the P2Ps are not secure, where various distinct nodes request to share their resources without preexistent trust relationships. Any small portion of malicious

DOI: 10.5121/ijnsa.2020.12201

members can interfere with data communication. Additionally, the fact that peers can join and leave without any centralized control, and at any moment raises a great security concern [1].

Consequently, the popularity of P2P applications among users utilizing a variety of these services has also enticed attackers to exploit the security flaws in P2P networks. Attacks aimed at P2P systems can be divided into three types. Attacks that target the overlay network layer, attacks that target the P2P routing protocols and attacks that launch on underlay P2P networks (Table 1).

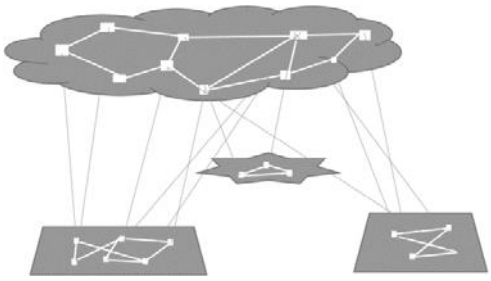

Figure 1. Overlay network

In this investigation, we first present the most well-known attacks and the type of vulnerabilities attackers attempt to exploit against P2P networks. Then defensive methods of these attacks will be explained respectively. Below the methods to securely communicate data in the presence of any malicious node have been stated. Section 2 of this chapter talks about various P2P networks that are susceptible to different forms of attacks. Security functions and operations are discussed in section 3. Attacks and their countermeasures are more deeply studied in Section 4. Section 5 presents the structure for secure communications. Section 6 analyzes the secured LDE-based structured P2P network. Section 7 gives conclusions about the current level of security in P2P systems.

2. PEER-TO-PEER OVERLAY NETWORK

Peer-to-Peer (P2P) is a virtual network that constructs on top of an underlying network (Figure 1). P2P networks are able to utilize distributed computational and data resources and provide direct resource sharing capability among peers. Their distributed construction provides scalability, efficiency and fault tolerance as core concepts. A P2P network is dynamic, self- organized and more importantly decentralized. The decentralization makes them prone to the single point of failure issue. Besides, they are cost-efficient since there is no need of servers and data centers. P2Ps scale organically with making any user a member for the design and this characterization makes them an environmentally friendly technology. Generally, there exist two categories of P2P networks: unstructured and structured P2P overlays. An unstructured P2P network [2] is constructed with some loose protocols, without any earlier awareness of the topology. In their early version, the overlay network practices controlled flooding as the instrument to direct queries across the network. Once a peer obtains the flooding query, it sends a list of all resources corresponding to the query of the sending peer. Although flooding centered methods are effective for finding vastly replicated resources, they are ineffective for discovering infrequent items. Evidently, this method is not scalable as each peer's load increases linearly with the total quantity of queries and the network size. Therefore, unstructured P2P overlay networks encounter one basic problem; nodes quickly become burdened, and hence the system does not balance when

processing a high degree of queries and unexpected increases in system size. Gunettela [3] and Yappers [4] are two examples of unstructured P2P architectures.

In newer versions of unstructured P2P, a few other methods have been introduced to reduce the impact of flooding.

- **Expanded ring search**: In this method, the querying node issues a sequence of queries initially with the small number of hops, if no reply is received, then it increases the hop limits.

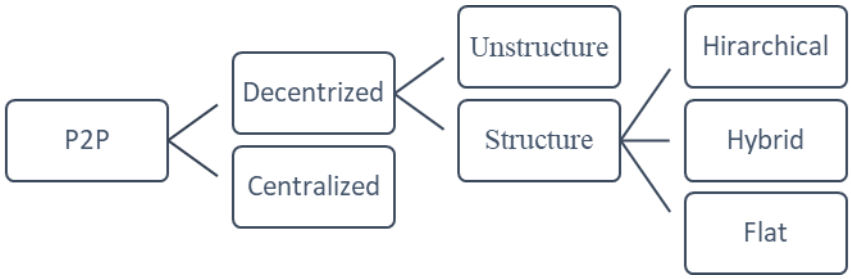

Figure 2. Hierarchy of P2P networks

- **Random Walk**: In random walk, the query propagates randomly throughout the network.
- **Gossiping**: In this approach, a node issues a lookup query to a neighbor who once it received a packet from it. A neighbor is also sending the request out to other neighbors in a similar manner. This method is comparable to spreading a virus in a community. Sometimes these approaches are called epidemic protocol as well.

To join an unstructured network, a new node primarily joins to one of several identified hosts that are usually available. Unstructured networks handle effectively the problem of churn. As it has been stated before, churn (peer connection and departure of the system) is frequent. Resource lookup-time complexity in a flat unstructured P2P network is *O* (n), where '*n'* is the number of peers in the P2P system. A properly designed structured architecture provides flexible facilities. In structured overlay networks, peers are organized into specific topologies. Typically, they utilize distributed hash tables (DHTs). The distributed hash table is a decentralized system of hash tables. DHTs are utilized to map resources to an identifier. As a result, it provides functionalities such as data lookup, insertion, deletion, etc., to the system.

By taking advantage of DHTs, a better complexity of *O* (log n) is achievable in contrast with unstructured networks. Nevertheless, maintaining DHTs is a complicated job and handling the problem of churn requires a huge effort. Hence, the key issue in such networks is to moderate the amount of work for churn processing while still delivering an effective data query service. Chord [5], Tapestry [6] and Pastry [7][8] are some noted structured P2P.

Additionally, few literatures has considered server-based P2P systems as one category of P2P architecture. They classified P2P networks into two main groups, centralized and decentralized, built on the existent of a server. Centralized P2P systems are a hybrid of client-server and P2P models. Usually, they have one or more servers to coordinates their peer resources. To find a specific resource, a message is sent to the system server by a requesting peer.

The server replies by sending the address of the resource holder. However centralized P2P applications are vulnerable to a single point of failure and they have limited scalability. Napster is the most famous example of centralized P2Ps. Decentralized P2P systems are separated into structured and unstructured classes. Furthermore, structured P2P can be categorized in different classes such as, Flat, Hierarchical, and Hybrid (Figure 2).

In Flat or single-tier P2P architecture, all nodes are only present in one overlay and all functions like routing are performed in that single overlay. It turns out that most of the overlays such as Chord, Pastry, and Kademlia [9] are flat. Hierarchical architecture (Figure 2) is a P2P overlay consisting of more than one structured overlay. They have different routing mechanisms that are built into their different layers. Generally, routing in one-layer leads to a gateway to another layer. Nodes are grouped into different clusters. Some nodes are in one overlay, and some others are participating in more than one. The nodes that are members of more than one layer with some special responsibilities are called Supernodes [10]. Supernodes have a large number of neighbors. They carry out tasks such as handling data flow and connecting the layers together.

Normally, supernodes are the computers with better bandwidth and stronger processor power in comparison to the other nodes. In addition, they have a longer on-line time. One of the best P2P overlay examples for hierarchical architecture is the LDE-based hierarchical P2P that is presented in Section 6. Hybrid P2P overlay is an architecture consisting of both structured and unstructured P2P and is embedded in different layers. HP2P [11], KaZaA [12], Gnutella 6.0 [13] are few examples of this type of architecture

3. SECURITY FUNCTIONS AND OPERATIONS

The basic protocols for P2P overlay networks are initiated on the notion that every node joining the system is trustworthy. However, in order to have a relatively secure P2P, this practice needs to be reconstructed according to the following security functions and operations.

Joining: A network user enters the group of P2P using a specific P2P program. After this, the user is enabled to interact and exchange resources with other participants. The process of joining must be inclusive of a mechanism that ensures authentication of every user. It is the admin of the group's security domain who will ensure this is done.

Leave: There is a leave function that allows a user to exit from the group. This means that they will not have to inform the domain administrator, or other peers for their reasons of leaving. The leave operation needs to implement a security domain such as an administrator policy regarding the leaving user's private data.

Search: Once a query is relayed by a user, other participants can receive and respond to it. Not all peers will, however receive the query. Hence, those who are unable to view the query can obtain it from the other peers. In order to ensure security, it is provided that only authenticated peers will relay queries in a network. This function is ensured by the security domain administrator.

Chat: It is possible for any peer to select a connected peer and directly communicate with them. The communication is conducted through text or voice messages. It is also the responsibility of the Security Domain Administrator (SDA) to ensure that only authentic participants send and receive messages in the group.

Routing: This is an operation in both the structured and unstructured P2P networks. Routing mechanisms should be structured such that only authenticated peers can participate in their operations.

Insertion and retrieval: Whenever a new resource is inserted, it is thoroughly assessed by peers from the routing table. Once credibility is determined, updates are done on the existing peer tables. SDAs have to ensure insertion and retrieval can only be done by participants who are authentic.

Update and Delete: It is the responsibility of the SDA to ensure that only modification of resource contents can only be done by authenticated through the operations of update and deletion.

Multicasting: This means that participants can pass packets of multiple tasks to each other. This, however can only be done by peers who are authenticated by the SDA.

4. ATTACKS ON P2P NETWORK

Creation of a decentralized peer to peer network was intended to equally spread services among participants. This, however, led to the emergence of security risks. Table 1 illustrates the classes of potential P2P attacks.

Table 1: Types of attacks, and their examples

Types of attack	Attack example
Attacks on Overlay P2P Network	• Pollution • Forgery • Omission • Man-in-Middle • Content Verification • Stealing Identity or Data
Attacks on P2P Routing Protocols	• Eclipse • Sybil • Churn • ID mapping • Poisoning the network
Attacks on Underlay Network	• DDoS • Query Flooding Attack • TCP SYN flooding attack

4.1. Attacks on Overlay Network

Pollution attack: Pollution attack occurs when the attacker deliberately inserts junk data and sends unusable information into another peer. The recipient peer may forward the spam and pollute the network in an exponential manner. This attack decreases the quality of service and potentially wastes the network bandwidth. Reputation-based techniques [14] are the strongest

method to fight against the pollution attacks as well as blacklisting, and utilizing hash verification and encryption.

Forgery attack: In this attack, the attacker tampers the data and damages its integrity and confidentiality. Clearly, message digest verification along with asymmetric keys for signing the message and symmetric cryptography for preserving confidentiality can be utilized to prevent this type of attacks [15].

Repudiation attack: Repudiation attack occurs when a peer denies receiving a message. Yet again, cryptographic procedures as message signatures are the best techniques to answer this vulnerability [15].

Omission attack: When a peer fails to forward the stream of data to other peers. In this attack, identifying the malicious node is very hard. These omission activities may even compromise the P2P network. Authenticate the peers by signed acknowledgments alongside monitoring and blacklisting the particular malicious peer was demonstrated to be an answer to discover the particular attacker [15].

Man-in-the-middle attack: The attacker in a man-in-the-middle attack intercepting between two nodes communications; acts like both sides and achieves access to data that the two parties were planning to exchange.

A central trusted authority, which commonly does not exist in most of the P2P networks, is required to authenticate each peer. Additionally, implementing an encryption mechanism to protect the confidentiality of the exchanging messages is another strong practice to detect and escape a man-in-the-middle attack [16].

Content Verification: Checking the validity of a document that is received may not be apparent as an actual attack. However, in a decentralized system, while there is no guarantee that peers are distributing the resource they assure. In fact, neglecting to verify the received content has been causing to spread malware throughout P2P networks [17]. Once again, reputation-based techniques [18] are the strongest method to fight against the pollution attacks as well as blacklisting, and utilizing hash verification and encryption.

Stealing Identity-Data: P2P systems are commonly utilized by users with constrained information about PC security. This hence makes their PCs and records to face heightened risks to attacks from advanced. Attackers can abuse requests of the inexperienced to help them gain access to sensitive information. In most instances, attacks will cause a leakage of the whole framework, passwords, or any other sensitive information.

4.2. Attacks on P2P Routing Protocols

Sybil Attack: The Sybil attack is probably the most challenging and difficult problem to solve in decentralized P2P networks. Douceur has first described this attack in 2002 [18]. In a Sybil attack, a sole malicious node produces many fake node identities and acts as various, individual physical peer in the system. These counterfeiting identities are called Sybils. In other words, an attacker tries to get a large number of fake peer-ids in a Distributed hash Table (DHT); consequently, it would be able to send false information and take control of the network substantially. For example, if an adversary can choose its identifier arbitrary, it can allocate itself

a collection of identifiers closer to certain resources in P2P overlay. In most of the structured P2P networks, nodes pick random IDs when they enter the network and the P2P system does not monitor the ID assignments, which allows an attacker to obtain as many as IDs it desires and use them to compromise the network. However, Linear Diophantine Equation (LDE)-based P2P limits the number of logical addresses per single resource type [19]. This technique prevents users from acquiring various fake identities. The LDE-based P2P is studied in detail in Section 6.

Sybil attacks can be prevented using various approaches. First and foremost, the number of user identifiers per single physical member must be limited. CAN [20] and Pastry generates the hash identifier, using the IP address of the peer. This technique is not reliable as the adversary is able to spoof many IP addresses to acquire more identities. Peer authentication in the network by using a logical ID is an effective mechanism against Sybil attack that has been integrated into LDE-based P2P [21]. Utilizing public and private keys along with a trusted, local key distribution center in LDE-based P2P makes it immune to any malicious peer that requests to obtain multiple fake identities.

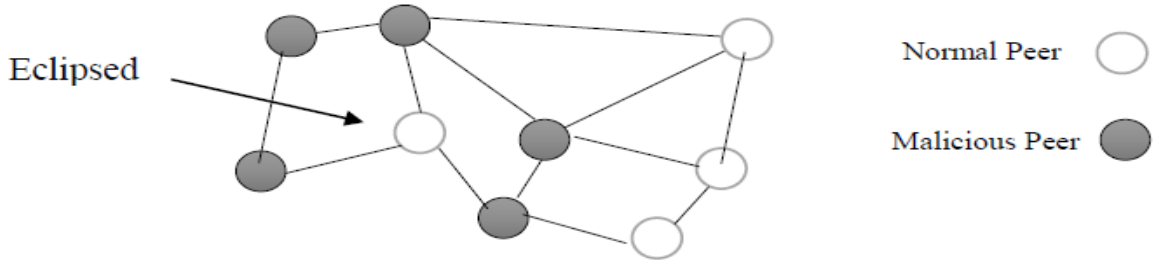

Figure 3. Eclipsed Attack

Eclipse attack: In DHT-based P2P networks each node in the network updates its own local and relatively small routing table. Their routing tables contain the addresses to a set of logical nodes. When a peer is performing a data lookup search in its P2P network, in fact, it attempts to resolve the IP address of the receiving node. Furthermore, it sends a request message to the destination peer that possesses the resource. If the destination node is not in its local routing table, the source node would perform a lookup request to any of its known peers. This peer tries to find a set of identifiers that are logically close to the destination peer. At some point throughout the data lookup,

Churn Attack: Another important attack that surprisingly has not been investigated much is churn attack. Churn is the main challenge in every P2P networks due to the impulsiveness of peers. Peers are able to join and leave the system constantly. The frequent joining and departing of the peers, which is identified as churn, arises vulnerability in P2P networks. Maintaining the routing tables due to the independent joining and leaving of peers can be very time and resource consuming. Consequently, as the rate of churn increases the P2P system as a whole may encounter serious issues and unable to respond to any join or lookup query. Therefore, an adversary could take advantage of the churn effect by producing thousands of nodes' arrival and departure in short cycles and weaken the P2P infrastructure. LDE-based P2P presents a solution to cope with the churn effect which is discussed in Section 6.

Poisoning the network: This basically is an approach of introducing useless data into the framework of a P2P hence poisoning it. Due to the need for P2P systems to actualize a query

administration in a DHT or a unified registry, it becomes possible to infuse numerous sets of queries into the file hence hampering the system. These false sets of inquiries will affect the duration of responses and may also lead to results that are not valid.

- *Index Poisoning*: Counterfeit data is implanted on lists that indicate an IP's objective and the number of the port. When a user tries to access data, the harmed file's counterfeit data is issued. The user will then make an association with the objective, and if the objective acknowledges the association, a TCP- association DDoS occurs.

- Routing Table Poisoning: Here, the attacker exploits the need of all P2P peers to keep up some sort of directing condition of the associated present peers. In a DHT framework, for example, every peer's table will have a specified number of neighbors given by the same number of system nodes. An approach used by attackers is the creation of fake neighbors in the routing table of all users. This results the target experiencing a surge in connection requests. In many occasions, P2P conventions will have an instrument used to expel stale peers from the routing table. After identifying the source of traffic, the objective is expelled from the routing tables of associating peers.

4.3. Attacks on Underlay Network

Distributed denial-of-Service (DDOS): This is an attack that can lead an entire network or assets contained in the network to be inaccessible to clients. These attacks can be explained as activities of gathering network node dispatches so that they can be used against the same victim. Attackers can utilize the questioning idea of P2P systems to over-burden the system and consequently handicap the entire network [22]. In the event a framework has a great number of users at a single time, a danger of filling in as a DDoS motive of launching an attack against a directed host is created. DDoS attacks have become significant in P2P frameworks because any node can go about as a router. For example, it is possible for a malevolent node at the centre of the system to effectively go about as a router and divert queries hence over-burdening another node identified as a target. Nodes with ill intent can cause harm through mishandling functions of the system. In many instances, this causes total network collapse [23].

It is also possible for assaults to exploit P2Ps to cause harm to other sites. Such attacks are usually instigated in order to exhaust resources and eliminate the limits of the objective. Some of the exhaustible assets include the processing power of an objective's CPU and its bandwidth. Other attacks on these setups involve injecting useless data into the system.

Malware: Numerous attacks, including those portrayed in this study, can be perpetrated by malware. For instance, content verification can be destabilized using malware. Perhaps the most significant malware family that exploits the P2P exchanges is the TDL4 malware [24]. The malware (TDL4), is widely known for being exceedingly diligent and difficult to expel from PCs. The reason for this is that, it affects the Master Boot Record (MBR). This is a segment of the hard drive which contains code executed amid the boot procedure, before the working framework begins. The attack is commonly used to appropriate other malware as a feature of pay-per-introduce plans, and cybercriminal partner programs.

Query Flooding Attack: This attack can occur in unstructured P2P such as Gnutella. In unstructured P2P networks, for a lookup, the user forms a query including the search keyword

and floods it out to its neighbors. Recipient peers match the keyword with their resources, if they find a match, query response messages are sent back to the sender containing information on how to download the resource. The peer that requested the file downloads it directly. In this attack, malicious nodes (which can be many peers due to Sybil effect) generate queries and flood the network as much possible.

TCP SYN Flooding Attack: This attack occurs when the TCP layer is flooded by 3-way hand-shake requests. This attack is a kind of Denial-of-Service attack and can aim any peer of the system or even on a larger scale can target the P2P network. In this attack, the adversary attempts to exhaust the system resources.

5. STRUCTURE FOR SECURE COMMUNICATION

User authentication: This is an approach through which systems verify whether a user is who they claim to be. Authentication is usually harder in P2P network architecture compared to centralized networks. This is brought about by the absence of a central server which can help determine a peer's identity. The only way to implement authentication is to apply the function within the system and between each pair or group of peers. Additionally, authentication can be applied in P2P networks by employing public/private keys. Identification and verification are two important components of authentication.

Privacy protection: Anonymity is a feature in which disables the revelation of identities. Some of the protocols of anonymity include Crowds, Hordes, and Freenet [22].

Data integrity: This is a function that makes sure that no modification, hampering, or loss of data occurs without proper authority. Because some data compromises can simply be duplicated computations, data should always be double-checked to verify if results are sensitive to the hardware setting of the machine. Another method of ensuring data integrity is to conduct tests on the peer machine so that the results machine is prevented from being used to conduct harmful computations.

Access control: These are mechanisms and policies which prevent access to computer resources. Sandbox to ensure this because of its ability to prevent access to non-shared resources.

Usability: The interface should be user friendly so that the information displayed can be easily understood. Additionally, controls can be provided so that users can change the settings in accordance with their preferences. Complicated interfaces cause unwanted changes to be made hence making it a threat to the network.

Availability: Data should be available to the authorized user when he or she wants access to it.

6. SECURED LDE-BASED STRUCTURE P2P NETWORK

Recently, a new hierarchical P2P network architecture has been introduced [25]. This is a fully hierarchical structured P2P system. To form the structure, and as the logical base of the hierarchical network, the Linear Diophantine Equation (LDE) has been utilized. It is to be observed that that most well-known structured architectures are based on use Distributed Hash Tables. To our best of knowledge utilizing LDE for creating P2P networks is a novel idea. The

various promising benefits that can be derived from using Linear Diophantine Equations have been found out.

In this architecture, we have designed two efficient data lookup algorithms: one for intra-group lookup query and the other for inter-group lookup query. The first one works inside a cluster while the second one involves more than one cluster. In our LDE-based P2P, we have incorporated strong security in the data lookup algorithms which is essential for any P2P system even though it is absent in many existing works.

In our work, any resources is denoted as a tuple $<R_i, V>$, where R_i signifies the kind of a resource and the resource's value is denoted by V. Any resource is capable of storing multiple values. For instance, if we consider R_i as the type of the resource 'books' and V' represent a specific book. Hence $<R_i, V'>$ signifies books (some or all) written by a specific author V'. In our fully structured hierarchical interest-based P2P network, we consider that the resource types are distinct. In other words, no two peers exist with the same value and resource. If S denotes the group of all nodes in a P2P network. Consequently $S = \{P^{Ri}\}, 0 \leq i \leq r-1$. At this point P^{Ri} symbolizes the subset comprising of all nodes that possess the same resource type R_i and none of the peers in P^{Ri} contain the identical value for R_i. We consider r as the number of individual resource categories exist in the network. Furthermore P_i in each P^{Ri} subset denotes the primary peer that join the system. Next, the P2P architecture design for LDE-based overlay system is explained.

6.1. Two Level Hierarchy

We present an overlay structured P2P architecture with two layers. Below is the detailed information of the network.

a. Level One (Figure 4): At this level, peers P_i $(0 \leq d \leq r-1)$ are forming a ring network. The number of existing distinctive resource types are denoted by r, which is also presents the number of nodes on the ring. The efficient data lookup is guaranteed by utilizing the ring network. The level one network is also called transit ring network.
b. Level Two (Figure 4): At this level, r numbers of fully linked networks of peers are existed. Respectively such cluster, G_i, is made by the nodes of the subset P^{Ri}, $(0 \leq i \leq d-1)$, in such a way that logically, all peers (ϵP^{Ri}) are directly connected to themselves, consequential in the network distance of 1. Peer P_i connects each of the level 2 networks (G_i) to the level 1(transit ring). We designation such a peer P_i as the group-head of network G_i.
c. Group-head peers in the level 1(transit network) are maintain a special table by the name of Global Resource Table (GRT). GRT contains of tuples of the form, *<Resource Type, Resource Code, Group Head Logical Address>*. The logical address allocated to a peer is called Group Head Logical in our P2P networks.
d. Any communication between a node $p_i \in G_i$ and $p_j \in G_j$ occurs exclusively via the corresponding group-heads P_i and P_j. The suggested architecture is illustrated in Figure 4.
   As it has been explained before, to calculate the logical addresses and form the architecture design, we utilize the products of a given Linear Diophantine Equation (LDE). The answers are used to define the subsequent.

a. Logical addresses of peers in a subnet P^{Ri} (i.e. group G_i). Use of these addresses will be

exposed to validate that all nodes in Gi are logically linked to each other creating a network of distance 1. Which means, each Gi is forming a complete graph.

b. Recognizing peers that are neighbors to each other on the level 1(transit ring network).
c. Representing distinct resource type's code, which being stored in GRT table.

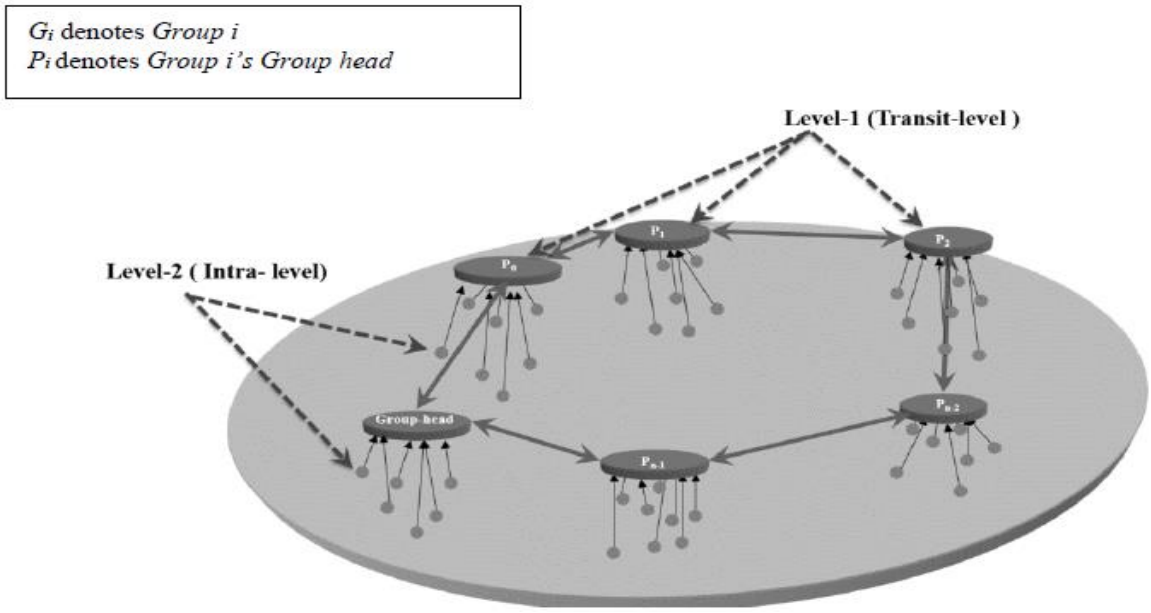

Figure 4. A structured hierarchical P2P architecture with distinctive resource types

6.2. Linear Diophantine Equation Based P2P Structure Design

Suppose that r different resource types exist in LDE-based hierarchical P2P network with maximum capacity of d resource types ($r \leq d$). Observe that d can be initialized to any value.

In each subgroup P^{Ri} (cluster G_i) node P_i acts as the primary node that joins the system and claims R_i as the resource type. At this point, to form the architecture the solutions of a specified LDE is used.

At level-1 the ring network (Figure 4) will contains of all *Pi*'s, in the following way:

a) Logical address $(n_0 + i.c/d)$ is given to each P_i.
b) Now because of modulo operation, the network connecting different clusters is a ring. If two peers' logical addresses differ by c/d, they are neighbors on the ring, the only exclusion is the first peer P_0 and the last peer P_{r-1}.
c) The ring network maximum diameter is $d/2$.

All peers at level-2 possessing one type of resource type R_i will make the group G_i (i.e. the subset P^{Ri}). Level-1 and level-2 are connected through group-heads only. Note that any message between any two groups G_i and G_j occurs by the particular group-heads P_i and P_j. Nodes in G_i will be given the logical addresses:

$$[(n_0 + i.c/d) + m.c], \; m \text{ is an integer}$$

Moreover for a given I, all addresses are mutually congruent solutions in G_i. Also as we know, the congruence relations are transitive, symmetric, and reflexive. Consequently, it can be determined that in a group G_i all peers are logically linked to each other and therefore making a network of diameter 1.

6.3. Complexity Evaluation

In following table, the lookup complexity of LDE-based approach has been stated, and been compared to some noted DHT-based approaches.

Table 2: Lookup Computational Complexity Evaluation [25] [26]

	CAN	**Chord**	**Pastry**	**LDE-based**
Structure	**Uses Distributed Hash Table**	**Uses Distributed Hash Table**	**Uses Distributed Hash Table**	**Uses Linear Diophantine Equation**
Lookup Procedure	Pair of{Key, value}	Corresponding PeerID, key	Corresponding key, prefix in PeerID	**Level-1 Sending via Group-heads Level 2: Direct connection**
Factors	N: total nodes in system d: dimensions	N: total nodes in system	N: total nodes in system b b: bits; B = 2	**r: distinctive types of resource** **N: total nodes in system** **r is very small compared to N**
Lookup Complexity	**1/d** ***O*(d N)**	***O*(log N)**	***O*(log N)** **B**	**Level-1: *O*(r)** **Level-2: Constant**

6.4. Secured Data Lookup

To achieve security from the viewpoints of authentication and confidentiality, we apply symmetric cryptography for intra-group data communication and asymmetric cryptography for inter-group communication. Symmetric key technique uses the same key for ciphering and deciphering. In symmetric cryptography, generating strong keys for the ciphers are relatively easier compared to its asymmetric counterpart. The encryption and decryption computations are faster since we use one key for both operations. Also, in general it is more difficult to break symmetric keys compared to asymmetric keys. However, it requires a secure way to distribute the shared keys among the peers. In our P2P architecture use of symmetric keys for intra-group communication appears to be suitable since all peers in a group form a complete graph and hence they all are one hop away from the group-head and from each other. In our system, we assume that group-heads are trustworthy peers and they act as trusted key distributed centers. Also, when a group-head crashes or leaves, the new group-head acts as a trusted center as well. However, for

inter-group communication, we take advantage of asymmetric cryptography. In asymmetric cryptography, the keys are not identical. For each secure communication, there is a pair of keys for encoding and decoding interchangeably. The key in the pair that can be shared openly is called the public key. The matching key, which is kept secret, is called the private key. Both keys can be used to encrypt a message; the other key can act in reverse [27].

Furthermore, to be able to support the use of asymmetric cryptography, we do a minor modification of Global Resource Table (GRT). A new entry is used in the GRT to represent the public key of each group-head. Group-head G_0 is responsible for updating the GRTs to reflect the effect of churn caused by group-heads leaving/joining the P2P system. Also, we assume that in each group, its members share a unique master key each with the group-head for secure intra-group communication.

6.4.1. Secured Intra-Group data Lookup

Regarding Intra-Group data look up, we consider that in group G_i, peer p_a possesses $<R_i,V_a>$ and requests for resource $<R_i,V_b>$. Notation K_{mn} denotes the master key shared only by a peer p_n ($\in G_m$) and the corresponding group-head P_m of group G_m. Thus, p_a has the master key, K_{ia}, known only to itself and the group-head P_i. For secure intra-group data lookup the following steps are followed in Figure 5.

1. p_a issues an encrypted request for resource $<R_i,V_b>$ to the group-head P_i.
   // this requested message is encrypted by the shared key K_{ia} of P_i and p_a.
   // thus, P_i is the only one who can successfully read the message and P_i knows
   //that it has originated at peer p_a
2. Group-head P_i decrypts the message with K_{ia}
3. Group-head P_i broadcasts in G_i for $<R_i,V_b>$
4. If peer p_b possesses $<R_i,V_b>$, it encrypts $<R_i,V_b>$ with K_{ib} and sends it to P_i
5. P_i decrypts the message with K_{ib}
6. P_i encrypts the message $<R_i,V_b>$ with K_{ia} and sends it to the requesting peer p_a
7. p_a decrypts the received message with K_{ia} and now has the resource $<R_i,V_b>$

Figure 5: Algorithm 1. Secured Intra-Group-Lookup

6.4.2. Secured Inter-Group Data Lookup

In our architecture, as we have discussed before, any communication between two peers p_i ($\in G_i$) and p_j ($\in G_j$). We use the notations Pu_m and Pr_m to denote respectively the public and private keys of group-head P_m. with no loss of generality, suppose a peer $p_i \in G_i$ demanding a resource $<R_j,V^*>$. Peer p_i recognizes that $R_j \notin G_i$. Suppose that there exist r distinctive resource types and $r \leq d$. The steps in Algorithm 2 are executed to answer the query (Figure. 6)

1. Peer p_i (ϵ G_i) encrypts the request for $<R_j,V^*>$ with K_{ii}
2. P_i dycrypts the message with K_{ii} and finds group-head P_j's address code from the Global Resource Table // *address code* of $P_j = n_0 + j\,(c/d)$
3. P_i calculates $h \leftarrow | (n_0 + i\,(c/d)) - (n_0 + j\,(c/d)) |$
   if $h > r/2$ then
   P_i encrypts the message with Pu_j and forwards the request to its predecessor P_{i-1}
4. **else**
   P_i encrypts the message with Pu_j and forwards the request to its successor P_{i+1}
5. **end**
6. Each middle group-head P_k forwards the requested resource until it reaches at P_j
7. P_j decrypts the message with its own private key Pr_j
8. **if** P_j possesses $<R_j,V^*>$
9. P_j encrypts the message with the public key Pu_i of P_i and unicasts it to P_i
10. **else**
11. P_j broadcasts the request for $<R_j,V^*>$ in group G_j
12. **if** $\exists$ p_k (ϵ G_i) which possesses $<R_j,V^*>$
13. p_k encrypts the request message with K_{jk}
14. P_j decrypts the message with K_{jk}
15. P_j encrypts the decrypted message with the public key Pu_i of P_i and sends it to P_i
16. P_i decrypts the message with its own private key Pr_i
17. P_i encrypts the message $<R_i,V_b>$ with K_{ii} and sends it to the requesting peer p_i
18. p_i decrypts the received message with K_{ii}
19. **else**
20. P_j unicasts *'search failed'* to p_i
21. **end**
22. **end**

Figure 6: Algorithm 2: Secured Inter-Group Lookup

7. CONCLUSIONS

Currently most existing structured P2P architectures use distributed hash tables (DHT) to form their networks. Use of DHTs guarantee efficient data insertion and data lookup operations in structured P2P systems. However, DHT-based architecture have been revealed to be extremely vulnerable to shield against security attacks. As it has been discussed, several investigations have been exposed their security issues. To resolve this shortcoming of DHT-based architecture, we have introduced a secured structured P2P network, which is based on number theory mathematical model, known as 'Linear Diophantine Equation (LDE) and its Mutually Incongruent Solutions' to realize the secured proposed architecture.

We have presented efficient LDE-based algorithms that provide secure data communications from the perspectives of confidentiality, authentication, and integrity. Additionally the evaluation results have shown that, the complexity of data lookup algorithms of the presented LDE-based P2P architecture outperforms DHT-based approaches.

REFERENCES

[1] Rahimi, S. (2017). A Novel Linear Diophantine Equation-Based Low Diameter Structured Peer-To-Peer Network. PhD thesis, Southern Illinois University, Carbondale, IL, USA.

[2] Lua, E. K., Crowcroft, J., Pias, M., Sharma, R., & Lim, S. (2005). A survey and comparison of peer-to-peer overlay network schemes. IEEE Communications Surveys & Tutorials, 7(2), 72-93.

[3] Ripeanu, M. (2001, August). Peer-to-peer architecture case study: Gnutella network. In Peer-to-Peer Computing, 2001. Proceedings. First International Conference on (pp. 99-100). IEEE.

[4] P. Ganesan, Q.Sun, and H. Garcia-Molina, "Yappers: A peer-to-peer lookup service over arbitrary topology," in Proceedings of the IEEE Infocom 2003, San Francisco, USA, March 30 - April 1 2003.

[5] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., & Balakrishnan, H. (2001). Chord: A scalable peer-to-peer lookup service for internet applications. ACM SIGCOMM Computer Communication Review, 31(4), 149-160.

[6] Ball, P., & Borley, N. R. (1999). The self-made tapestry: pattern formation in nature (Vol. 198). Oxford: Oxford University Press.

[7] Rowstron, A., & Druschel, P. (2001, November). Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing (pp. 329-350). Springer, Berlin, Heidelberg

[8] Hoßfeld, T., Oechsner, S., Tutschku, K., Andersen, F. U., & Caviglione, L. (2006, March). Supporting vertical handover by using a pastry peer-to-peer overlay network. In Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on (pp. 5-pp). IEEE.

[9] Maymounkov, P., & Mazieres, D. (2002, March). Kademlia: A peer-to-peer information system based on the xor metric. In International Workshop on Peer-to-Peer Systems (pp. 53-65). Springer, Berlin, Heidelberg.

[10] Liang, J., Kumar, R., Xi, Y., & Ross, K. W. (2005, March). Pollution in P2P file sharing systems. In Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies. (Vol. 2, pp. 1174-1185). IEEE.

[11] Peng, Z., Duan, Z., Qi, J. J., Cao, Y., & Lv, E. (2007, January). HP2P: A hybrid hierarchical P2P network. In Digital Society.

[12] Liang, J., Kumar, R., & Ross, K. W. (2004). Understanding KaZaA.

[13] Klingberg, T., & Manfredi, R. (2002). Gnutella 0.6. Network Working Group.

[14] Dhungel, P., Hei, X., Ross, K. W., & Saxena, N. (2007, August). The pollution attack in P2P live video streaming: measurement results and defenses. In Proceedings of the 2007 workshop on Peer-to-peer streaming and IP-TV (pp. 323-328). ACM.

[15] Mondal, A., & Kitsuregawa, M. (2006, September). Privacy, security and trust in p2p environments: A perspective. In 17th International Workshop on Database and Expert Systems Applications (DEXA'06) (pp. 682-686). IEEE.

[16] Yang, Y., & Yang, L. (2012). A survey of peer-to-peer attacks and counter attacks. In Proceedings of the International Conference on Security and Management (SAM) (p. 1). The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

[17] Washbourne, L. (2015). A survey of P2P Network security. arXiv preprint arXiv:1504.01358.

[18] Douceur, J. R. (2002, March). The sybil attack. In International workshop on peer-to-peer systems (pp. 251-260). Springer, Berlin, Heidelberg.

[19] Rahimi, N., Sinha, K., Gupta, B., Rahimi, S., & Debnath, N. C. (2016, July). LDEPTH: A low diameter hierarchical p2p network architecture. In 2016 IEEE 14th International Conference on Industrial Informatics (INDIN) (pp. 832-837). IEEE.

[20] Ratnasamy, S., Francis, P., Handley, M., Karp, R., & Shenker, S. (2001). A scalable content-addressable network (Vol. 31, No. 4, pp. 161-172). ACM.

[21] Rahimi, N., Gupta, B., & Rahimi, S. Secured Data Lookup in LDE Based Low Diameter Structured P2P Network. Proc. CATA 2018, March, Las Vegas.

[22] Saboori, E., & Mohammadi, S. (2012). Anonymous communication in peer-to-peer networks for providing more privacy and security. arXiv preprint arXiv:1208.3192.

[23] Naghizadeh, A., Berenjian, S., Meamari, E., & Atani, R. E. (2016). Structural-based tunneling: preserving mutual anonymity for circular P2P networks. International Journal of Communication Systems, 29(3), 602-619.

[24] Royal, P. (2012). Entrapment: Tricking malware with transparent, scalable malware analysis. Talk at Black Hat.

[25] Gupta, B., Rahimi, N., Hexmoor, H., Rahimi, S., Maddali, K., & Hu, G. (2018, July). Design of Very Efficient Lookup Algorithms for a Low Diameter Hierarchical Structured Peer-to-Peer Network. In 2018 IEEE 16th International Conference on Industrial Informatics (INDIN) (pp. 861- 868). IEEE.

[26] Gupta, B., Rahimi, N., Hexmoor, H., & Maddali, K. Design of a New Hierarchical Structured Peer-to-Peer Network Based On Chinese Remainder Theorem. Proceedings of the 33rd International Conference on Computers and Their Applications.

[27] Rahimi, N., Reed, J. J., & Gupta, B. (2018). On the Significance of Cryptography as a Service. Journal of Information Security, 9(04), 242.

AUTHOR

Dr. Nick Rahimi is a Cybersecurity assistant professor in Southern Illinois University (SIU). He is committed in researching blockchain, cryptography, peer-to-peer networks, software security, and Internet censorship. He earned his Ph.D. and M.S. in Computer Science from Southern Illinois University. Nick obtained two B.S. degrees in Computer Software Engineering and Information Systems and Technologies.