Security in P2P Networks Systems

Shailesh D Wagh

George Mason University
Fairfax, VA, USA

{swagh2}@masonlive.gmu.edu

***Abstract*— P2P applications are considered more vulnerable than client server applications. It's also been believed that it is much more difficult to implement the security measures successfully than other major counterparts. In this paper , I am trying to provide security measure for p2p applications against popular attack and seek to discuss the common challenges and best practices when building a secure P2P Network applications including: security, node churn, search, data persistence, and the necessary protocol stacks needed to deploy a P2P Network. I also discuss how traditional implementation methods can be improved by utilizing the existing social connections between each peer.**

Keywords—P2P, Distributed System, P2P Security, P2P protocols, Lookup .

I. INTRODUCTION

Internet is itself inherent decentralized and distributed and p2p applications are trying to mimic similar simpler architecture with highest efficiency in scaling. With the introduction of WebRTC supporting browser to browser video calling, file sharing etc, p2p application security issues will be under higher scrutiny. To understand the vulnerability and solutions suggested in the later part of the paper, it is necessary to understand the major basic architectures of p2p because different architectures have different vulnerability so different mitigating techniques need to be used.

II. ARCHITECTURES FOR P2P SOCIAL NETWORKS

A. Unstructured P2P Architectures

Unstructured P2P Architectures are distinguished by a lack of structure in its virtual overlay network, the lack of constraints on resource distribution, and minimal network growth policies [3]. Connections are made when nodes randomly join or exit the network. As a result, Unstructured P2P networks are highly robust when presented with the problem of "node churn" [3]. In addition, they are also the simplest to deploy.

Challenges in Unstructured P2P networks mainly deal with search algorithms. As a result of its lack of structure in its virtual network overlay, a search query must traverse a large amount of network before a result is found. Several different solutions have been proposed to solve this problem including: Flooding, Random Breadth-First-Search, Directed Breath-First-Search, k-Random Walk, and Generic Adaptive Probabilistic Search [4].

The most popular file sharing application which utilizes an Unstructured P2P architecture was Gnutella v0.4 [5]. In order to conduct a search, a user would flood the network with the request. Each peer which received the request would query their file system for the data and either respond to the request or forward the request to its neighboring peers. This implementation of search creates a large amount of overhead within the network. As a result, Gnutella adopted a Structured P2P architecture in v0.6. This architecture employed the use of leaf nodes and ultra-nodes. Also called as super nodes, ultra node use is not recommended because of high reliance on ultra nodes creating vulnerability for worm propagation.

B. Structured P2P Architectures

Structured P2P Architectures are categorized by the implementation of a structured overlay network, the constraint of network resources, and strict network growth policies [3]. Through these invariants, the network forms a specific graph structured which can provide efficient search queries.

The most common type of Structured P2P network is a distributed hash table (DHT). Its key feature is the use of a hashing function to assign ownership of a data file to a peer. Examples of DHT implementations include: Chord, Pastry, CAN, and Tapestry. Research has shown that a DHT search has upper bound of O(log n) when conducting a search [6].

An excellent example of this implementation is Gnutella v0.6 with its use of leaf nodes and ultra nodes. Leaf nodes typically connect to 3 peers while ultra nodes connect to more than 32 [5]. Gnutella search algorithm utilizes this relationship to traverse the network more efficiently and reduces network overhead. When a search is initiated the algorithm first searches peers in neighboring leaf nodes. If unsuccessful, it forwards the request to a higher tier.

Most challenges with Structured Architectures revolve around maintaining the invariants in the face of node churn. In order to conduct an efficient search, a node must maintain a list of neighbors as well as their relative attributes. In addition, special hash functions must be employed to maintain the key space and balance of each key-value pair [6]. I discuss solutions to these issues in section VII-C.

C. Hybrid Architectures

Hybrid Architectures maintain both a client-server network and a P2P network. Applications of this type of architecture can be found in file sharing and instant messaging applications. There are three main services that are provided: login, query, and download [7]. When a peer logs into the system, he does so through the normal client-server schema. When a request is initiated, the server retrieves the data by completing three tasks. First, the server searches the clients cache to see if the data has already been downloaded. Second, if the data does not exist in the client's cache, the server start to retrieve the data from the server. Finally, the server searches other nearby nodes for copies of the requested data. If found, it initiates a request for the data for the client.

III. Protocol Stacks for P2P Social Network Applications

A. Application Layer

UP2P SIP

UP2P SIP is a P2P version of the SIP framework. The protocol is designed for mobile Internet telephony. The UP2P SIP builds a P2P overlay in an unstructured fashion. Each node maintains a buddy list which can be used to monitor the presence of its friend nodes [8]. Also, neighbor tables are provided in order to facilitate connections. UP2P SIP works well in a small P2P networks because it utilizes a flood method for its search and discovery methods, however structured implementations are possible.

B. Transport Layer

JXTA

JXTA is an open source P2P protocol which bundles several different protocols together as a P2P middleware API. Included in this set is the Peer Discovery Protocol (PDP) used for advertising resources. Peer Resolver Protocol (PRP) is used when peers transmit queries to at least one target peer [12]. Peer Information Protocol (PIP) is a protocol used when a peer acquires status information of other peers [12]. A Peer Membership Protocol (PMP) is a protocol, which enables peers to organize themselves and perform grouping of peer groups. A Pipe Binding Protocol (PBP) is a protocol, which forms a virtual pipe or communication channel between one or more peers. A Peer Endpoint protocol (PEP) is a protocol which can find a route so that one peer can transmit a message to another peer [12].

IV. SECURITY IN P2P NETWORKS

A. Access Privileges in P2P

Access to selected information of user specific information or whole documents in P2P sharing should be controllable [15]. Access rights must be changeable at any time if the owner of the document decides to do so. However, there must be no restrictions on the owners that store the data. To achieve this, Access Control Lists(ACL) should be use over Capability Lists to improve efficiency.

Write and Read Access: A file which is to be shared should be created with a timestamp and signed by the user for later verification of the author. The user should also define which other users should be allowed to read this item. If the user decides that only a set of privileged users should be allowed to read the item, a symmetric key is created and file, including its signature, is encrypted with this key [15]. This symmetric key should then be encrypted with the public keys of the privileged users, which leads to n encrypted copies of the symmetric key for n privileged users [15]. The encrypted copies of the key are then attached to the file, which is then signed and finally stored as a CryptedItem in the network. Each file has an objectID, which indicates where in the DHT the file will be stored. Using this ID, the file can be retrieved by a privileged user.

Access Control in Groups: Access to documents in a group can be granted to all group members. Just one symmetric key is required for all accessible data inside a group. This symmetric key is created by the owner of the group at the time the group is created [15]. Also, a list of symmetric key is created where the symmetric key encrypted with the public key of the group members is stored [15]. For each new member that joins the group, the administrator just adds a copy of the symmetric key, encrypted with the pursuant public key. This list is stored in the network, signed with the administrator's private key to inhibit unauthorized write access. The objectID of this list is a hash of the administrator's public key and the name of the group [15]. A user can thus store new file just as described above with the only difference that if he/she wants to make the item only accessible to group members, it is encrypted with the symmetric key of the group.

B. Security Issues in P2P Networks

P2P networks are prone to many security issues due to the lack of a central authority which identifies peers and approves the content being shared. In a social media, it is easy to steal an identity and use it for malicious purposes. Such malicious peers may give incorrect responses to requests, and also send corrupt or harmful data to the receiver. This might lead to trust issues and wastage of resources and time for downloading content. Also, the receivers might also be attacked with virus and other threats. Below are some common attacks which the P2P networks might encounter.

Malicious peer groups: When a peer joins a P2P network, it might unknowingly join a malicious network which is a serious threat to the peer's data and identity [9].

Pseudo spoofing: Here, an attacker creates a large number of identities, thereby can control a substantial fraction of the system [14].

P2P Worms: P2P worms in social networks may use a propagation method by locating potential targets. Later on, these worms can cause DoS attacks and install/run malicious code remotely [9].

Folder sharing: Misconfigurations of receiving peer can cause unintentional sharing of its resources. This might lead to possible loss of data confidentiality, integrity if the malicious peers gain access [9].

Routing Attacks: Malicious users might also perform "routing attacks", or denial of service attacks. In a common routing attack called "Incorrect lookup routing", malicious nodes forward request incorrectly in order to return false results. “Incorrect routing updates" is an attack where a malicious node corrupts routing tables of neighboring nodes by sending them false information. “Incorrect routing network partition" is done when new nodes are joining, bootstrap is done via a malicious node and this places the new node in a network that is populated by other malicious nodes [9].

C. Pseudo Spoofing

Providing anonymity, robust access controls, data integrity, and confidentiality services for users of P2P networks is an increasing need. Fundamental security threats are malicious parties which are able to claim multiple identities. These attacks are called pseudo spoofing and without the foundation of stable, verifiable identities, it is difficult to provide a set of desired security services. Such services are important as the industry moves towards using P2P technology in applications and in e-commerce. The security situation for P2P networks is made worse because they lack any centralized authority who can verify identities of all peer systems in the network.

In a P2P environment, there are no trusted third parties who can provide assurances as to the identities of entities in the network [14]. Instead, each peer is identified based on a pseudonym that is selected by the peer for itself. In most P2P networks, there exists no standard registration methodology on these pseudonym selections, and indeed entities can claim multiple pseudonyms, including those of other entities in the network [14]. Pseudo spoofing refers to a peer creating and handling more than one pseudonym at once. A potential attacker might manipulate tens or hundreds of pseudonyms to his benefit. An attacker might also make use of pseudonym of an existing peer, preferably a peer with good reputation which is trustworthy [14]. This kind of attack is called pseudo theft. An attacker who engages in pseudo spoofing or pseudo theft in a P2P network may do so in order to manipulate a reputation mechanism in place in that network. The attacker can then perform illegal activities and can send corrupt data with this fake identity. However, a peer, having gained a bad reputation, can discard the corresponding pseudonym and re-join the network with a fresh pseudonym. In on-line auction systems, a peer can use a pseudonym to generate false bids, an attack known as shilling [14]. In general, a peer can create a number of pseudonyms, build up the reputation of one or more of their pseudonyms using the others, and then make use of the now reputable pseudonym(s) to persuade other peers to trust him [14]. Clearly any P2P network attempting to rely on a reputation system for building trust in peers would need to address the problems of pseudo spoofing and pseudo theft. As an example, the reputation system used in eBay, essentially a P2P system, has constantly been under pseudo spoofing attacks. It has been argued that in any P2P system without a centralized point of trust, such attacks on identity can never be effectively fought [14].

The approach of using pseudonyms is clearly in conflict with the requirement for stable identity, as I have noted above, to provide many security services [14] This problem of providing entity authentication in the face of un trusted networks of peers seems to me to be a central challenge in providing wider security services for P2P networks.

V. SOLUTIONS TO SECURITY ISSSUES

I am trying to explain the proposed mitigation techniques in the latest research along with currently used efficient techniques. Combination of implementation of multiple security measure will be used as usually multiple vulnerabilities are exploited at once to make a successful attack .

A. Divergent Lookup :

Prof. Neeraj Suri and Prof. Dr. Thorsten Strufe believe that the lookup mechanism of the peer is the one of the key vulnerabilities for many pseudo-spoofing attacks like Eclipse.

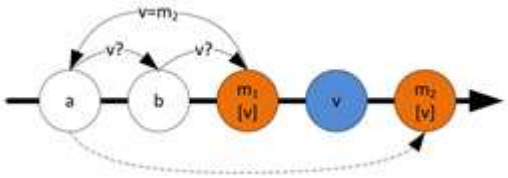


Fig. 1 Eclipse attack against overlay convergent lookup

Convergent lookup can be observed above where at most closest peers to the destination are picked from Queue i.e. node ‘a’ querying nearest node ‘b’ about node ‘v’. This primitive sends lookup request messages to other peers picked from the queue Q, requesting contact information for v. If queried peers do not know about v, these return closer contacts towards as a proposal for further requests in the next iteration. It can be easy to place a malicious node to attack a specific peer while carrying out man in the middle attack leading to successful eclipse attack.

New proposed lookup is divergent lookup- Random walks

1. Avoid destination’s proximity
2. Cancel convergent iterative lookup criteria

So now lookups do not converge towards the proximity threshold and potentially starve.

CPL is defined as common prefix length which is a distance metric between 2 nodes and t_p which defines the proximity threshold in terms of CPL depending on that overlay network. So, peers with CPL > t_p with k, are removed to avoid possible malicious peers in Q.
Peers are selected in a random fashion which replaces the convergent distance decreasing strategy. For the criteria of taLEA, authors have proved this change in lookup to be effective.

B. Directed Anonymous Attestation

Direct Anonymous Attestation (DAA) is a method to confront pseudo spoofing or pseudo theft. It can be used to enforce the use of stable, platform-dependent pseudonyms and thus reduce pseudo spoofing in P2P networks [14]. This algorithm runs on a protocol designed by Trusted Computing Group (TCG), referred to as a verifier in the algorithm.
In this algorithm, initially, the platform(peer) goes through a join phase where it obtains a credential from a trusted third party called an issuer. The platform is asked to become a member of the group of platforms to which credentials have been issued by that issuer [14]. The platform proves that it has knowledge of a specific secret value f and is authenticated to the issuer via its unique key called EK(Endorsement Key) [14]. If the issuer accepts this, it provides a credential in the form of a signature to the peer. This credential enables the platform to convince a verifier that it has previously successfully completed the join phase with a particular issuer [14]. Once the join phase is complete, the DAA-Signing algorithm can be used to prove to any verifier that it has an appropriate issuer credential and at the same time, sign a message m using the secret value f. This allows any shared data to be checked as being genuine by the verifier. Corresponding to the DAA-Signing algorithm is a DAA-Verify procedure that is executed by the verifier.

Throughout the algorithm, the platform is only identified by a pseudonym. The verifier usually helps in forming the pseudonym. By using digital signatures for verifying the identities of peers, this algorithm helps in eliminating pseudo spoofing attacks.

C. Social VPN

A Social VPN is a virtual private network created among peers, based on the relationships established between them through social networking service [17]. It provides peer-to-peer (P2P) network connectivity between a user and friends, by hiding the complexity in setting up, maintaining public, private keys, authenticated end-to-end VPN tunnels from the users.

Major functionalities such as querying a peer's list of connections and storing, retrieving peer's unique overlay identifier (keys) are performed by a social VPN [17]. The social VPN software at each endpoint keeps a list of current friends, which is periodically updated. The VPN software at sender's endpoint generates a host private key and uses it to self sign a host certificate [17]. For each new friend, the VPN automatically downloads their respective public key and places it in a repository accessible to the network tunneling stack. It then uses APIs to authenticate and publish the sender's public key to a data store hosted by the social networking site with the sender's identifier. The social networking site then, authenticates both peers using password authentication. Peers should use a software that authenticates the social networking site using SSL/TLS and a signed x.509 certificate (as is standard in web browsers) [17]. Thus, the communication is secured by authenticating both peers in a social VPN in P2P systems.

VI. IMPLEMENTATION METHODS

A. Flooding

Flooding is a popular algorithm to search files in an unstructured P2P network. Each node in the algorithm acts as both a sender and a receiver and tries to forward every search query to all its neighbors except the source node. Each search query has a unique number and duplicate queries with the same number are discarded to avoid redundancy [19]. For each query, flooding is performed in a hop by hop fashion counted by time-to-live (TTL) counter. Initially, a query's initial TTL set to a specified value, which is decremented by one when it travels between two nodes. A query comes to its end only when it becomes a redundant query, when its TTL is decreased to 0 or when the data it is looking for is found.

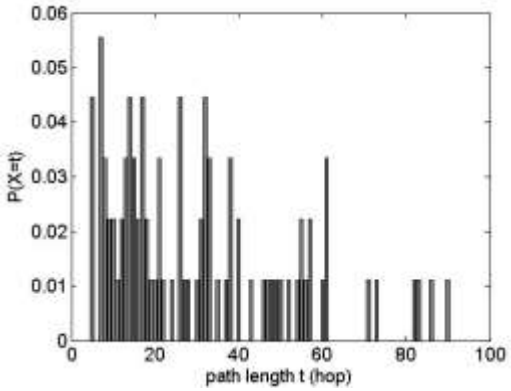
Queries are generally indiscriminately broadcasted in a whole neighborhood because of a lot of network resources [19]. As a result, its search efficiency decays as the search time increases since the number of query messages increases with the size of visited peers. The algorithm faces scalability problem when the query time increases [19]. However, as this algorithm demands very little management overhead and takes advantage of the spontaneous replication of popular content, it is widely used in popular Gnutella system.

B. Small Search Greedy Algorithm

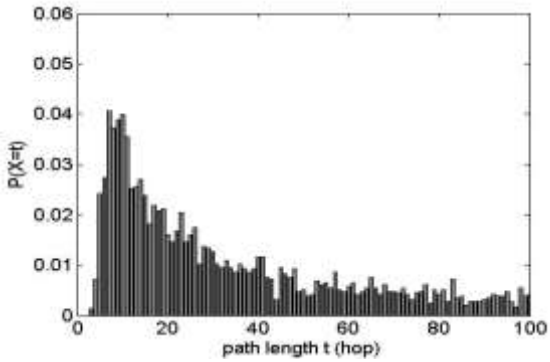
Many methods have been implemented to build a decentralized approach for efficient social routing in social P2P networks [13]. When compared to previous times, friendship connections in real-world social networks are too complex and hence, routing strategies based on one-dimensional social information are less efficient in real-world social networks [13]. Small search greedy algorithm solves this problem and provides an efficient method for routing. This algorithm helps in reaching random target nodes in real-world social networks by making a routing decision in each hop. This routing algorithm is based on a social distance metric comprising of two variables: geography and common interest.

In this algorithm, the geographic distance between each node and target is represented by the Euclidean distance between the geographic coordinates of the two nodes [13]. The interest based distance between each node and the target is the height of their lowest common ancestor in the interest

hierarchy. After these distances are measured, a normalization factor is used to normalize these two social distances as a ratio between 0 and 1 [13]. Once the normalized distances are calculated, the neighboring node closest to the target node in any of the dimensions is selected as the next node in the greedy routing scheme. This is continued until the target node is reached. Thus, by extending the greedy routing algorithm to support two dimensions, routing is enhanced by using a negligible amount of processing and storage.



(a) Geography Only Routing[13]



(b) Interest Only Routing[13]

Fig. 1.

Figure 1(a) clearly shows that the distribution of the path lengths(distance) of reaching target person where the majority of the path lengths are below 40 with a few a high as 90 hops. In figure 1(b), the distribution of path lengths in the experiment of reaching target person are more skewed around the median path length of 23 with a long tail up to the maximum TTL of 100. These two values depict the reduction of number of hops to reach targets nodes on employing this algorithm.

C. Reactive and Periodic Failure Recovery

The dynamics of peer participation, or churn, are an inherent property of P2P systems. Accurately characterizing churn requires precise and unbiased information about the arrival and departure of peers, which is challenging. Distributed hash tables which use a structured key based routing handle node churn failures through the below failure recovery mechanisms [18].

In reactive recovery mechanism, a DHT node tries to find a replacement neighbor immediately upon noticing that an existing neighbor has failed [18]. It sends a copy of its new neighbor list to every node in its neighbor set. Under low churn, reactive recovery is very efficient, as messages are only sent in response to actual change when failure is detected, whereas periodic recovery is wasteful.

In contrast, in periodic recovery a node periodically shares its set of neighbors with each of the members of that set. This process takes place independently if the node is detecting changes in its neighbor set [18]. As a simple optimization, a node picks one random member of its neighbor set to share state within each period.

There is a huge difference in the bandwidth consumed under different churn rates. Under low churn, reactive recovery is very efficient, as messages are only sent in response to actual changes, whereas periodic recovery is wasteful [18]. As churn increases, however, reactive recovery becomes more expensive. In this case, a node sees more failures when its neighbor set is larger, but the set of other nodes it must notify about the resulting changes in its own neighbor set is higher. Periodic recovery, however, aggregates all changes in each period into a single message.

D. Message Timeouts

Detecting failure in P2P is done through sending message timeouts. Also, a node should ensure that the timeout for a request is judiciously selected before routing to an alternate neighbor [18]. Nodes should accurately choose timeouts such that a late response is indicative of node failure, rather than network congestion or processor load. There are two timeout calculation strategies which are as follows:

TCP-style timeout: In a DHT, each node periodically probes each neighbor for availability and thus, a node can easily maintain a past history of each neighbor's response time [18]. This can be used in calculating timeouts.

Timeouts from virtual coordinate: In this scheme, timeouts are computed without previously measuring the response time to every node in the system [18]. A distributed machine learning algorithm is hence employed to assign to each node coordinates in a virtual metric space such that the distance between two nodes in the space is proportional to their latency in the underlying network.
Thus, based on these timeouts, recovery schemes are employed for 100% efficiency through social networking service in P2P network connectivity.

E. Proximity Neighbor Selection

Proximity neighbor selection (PNS) is a method where a DHT node, with a choice of neighbors, tries to select those that are most nearby itself in network latency. For this selection, several algorithms are employed. In this method, initially, one of the PNS algorithms is used to find nodes that may be near to the local node [20]. Next, the latency to those nodes is measured. If there are no existing neighbors in the routing table entry that the measured node would fill, or if it is closer than the existing routing table entry, we replace that entry, otherwise we leave the routing table unchanged [18].

Global Sampling: In global sampling, the lookup functionality of the DHT is used to find new neighbors. A lookup for a random identifier is done and the node returned by this lookup will almost always have the desired prefix [18].

Neighbors' neighbor: If a node discovers one nearby node, then that node's neighbors are probably also nearby. In this technique, we contact an existing routing table neighbor at level l of routing table and ask for its level l neighbors [18].

Neighbors' inverse neighbors: In this approach, instead of sampling neighbors' neighbors, those nodes which have the same neighbors as the local node are sampled. However, this method requires additional bandwidth because normally, a DHT node would not record the set of nodes that use it as a neighbor [18].

Recursive sampling: This process starts by sampling from the routing table, starting with the highest level in its routing table, where a node contacts the neighbors at that level and retrieves their neighbors. The latency to each newly discovered node is measured, and all but the k closest are discarded. The node then decrements l and continues the same for all nodes. Along the way, each discovered neighbor is considered as a candidate for use in the routing table [18].

F. Dunbar Approach

Social Networks are the storehouses of huge amount of data in the form of messages, photos and personal information. The main challenge of these networks is guaranteeing the data persistence of a user data even when he is offline [16]. Some DOSNs rely on external storage systems, while some other propose to store social data of a user on the storage support provided by users' friends. In these proposals, the data owner serves his own data when he is online, and elects a subset of his friends to make the data available when he is offline.

Dunbar approach is similar to the second approach mentioned above. Here, each user dynamically elects a subset of his friends, which we can call Points of Storage (PoS), to store a replica of his social data. The PoS of a user serves his data when he is offline [16]. Each node dynamically elects a minimal set of Point of Storages among his friends. Each PoS elects another node as PoS if it disconnects from the system, and this election may be recursively executed [16]. As long as there is at least one online friend for each node, the availability can be always satisfied but if no friend is online, then data stored in the system will not be accessible by any means. Also, if a friend of the user reconnects to the system and does not keep in memory a copy of the user's profile he cannot access the user's data until one PoS of the user or the user himself reconnects to the system. This permits us to reduce the total amount of social information each peer has to keep in memory and the number of connections of the P2P overlay [16]. Thus, all systems are connected to links and all the links among those guarantee that there is always a copy of the profile on an online node for each node in the network. Electing a trusted POS is a fundamental issue in this approach and if this exerted by using an effective algorithm, this approach guarantees the consistency of the copies of the users' data.

VII. CONCLUSION

It can be safe to say that choosing the right P2P architecture for various functionalities such as routing, security, and search are based on various aspects of service that need to be provided. It is difficult to implement p2p application with robust security compared to other. If the application requires a high degree of security, I recommend Hybrid P2P Network using SSL/TCP. If the application emphasizes speed of search for data files, I suggest a Structured P2P network using UDP. If the application provides telephone emulating services, a Structured P2P network using UP2P SIP is to be preferred. All applications with a small user base should use an Unstructured P2P (UDP or TCP). Furthermore, the use of Social VPN should be implemented whenever possible because it mitigates many of the security threats in P2P.

REFERENCES

[1] Government of Hong Kong SAR, "Peer-to-peer Networks", http://www.infosec.gov.hk/english/technical/files/peer.pdf,Febuary 2008.

[2] Wikipedia, https://en.wikipedia.org/wiki/Peer-to-peer, April 2016.

[3] Fletcher, Sheth, Boerner, "Unstructured Peer-to-Peer Networks: Topological Properties and Search Performance", G. Moro, S. Bergamaschi, and K. Aberer (Eds.), AP2PC 2004, LNAI 3601.

[4] Xiuqi Li, Jie Wu, "Searching Techniques in Peer-to-Peer Networks", https://courses.cs.washington.edu/courses/cse522/05au/searchingsurvey.pdf, April, 2016.

[5] Wikipedia, https://en.wikipedia.org/wiki/Gnutella, April 2016.

[6] Sameh El-Ansary, Seif Haridi, "An Overview of Structured P2P Overlay Networks", http://eprints.sics.se/237/1/elansary-singlespaced.pdf, April 2016.

[7] Beverly Yang, Hector Garcia-Molina, "Comparing Hybrid Peer-to-Peer Systems", Proceedings of the 27th VLDB Conference, Rome, Italy, 2001.

[8] Chien-Ming Cheng, Shiao-Li Tsao, and Jin-Chang Chou, "Unstructured Peer-to-Peer Session Initiation Protocol for Mobile Enviroment", The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07), 2007.

[9] Nguyen Hoang Anh, "Peer-to-Peer Systems: A Shared Social Network", TKK T-110.5190 Seminar on Internetworking, 2008-04-29/30.

[10] “http://www.siptutorial.net/SIP/example.html”, April 2016.

[11] “http://resources.infosecinstitute.com/udp-hole-punching/”, April 2016.

[12] Il-Woo Lee, Moon-Ok Choi, Ho-Jin Park, and Kwang-Roh Park, “Social Networking Service based on Peer-to-Peer Network”, The Third International Conference on Systems and Networks Communications, 978-0-7695-3371-1/08.

[13] Shuo Jia, Pierre St Juste, and Renato J. Figueiredo, “A Multidimensional Heuristic for Social Routing in Peer-to-Peer Networks,” The 10th Annual IEEE- CCNC Social Networking & Social Media Track, 2013.

[14] Shane Balfe, Amit D. Lakhani and Kenneth G. Paterson, “Trusted Computing: Providing Security for Peer-to-Peer Networks,” Fifth IEEE International Conference on P2P computing, 2005.

[15] Kalman Graffi, Patrick Mukherjee, Burkhard Menges, Daniel Hartung, Aleksandra Kovacevic, and Ralf Steinmetz, “Practical Security in P2P-based Social Networks,” IEEE 34th Conference on Local Computer Networks, 2009.

[16] R. Nicole, Andrea De Salve, Barbara Guidi, Francesco Pitto, and Laura Ricci, “Trusted Dynamic Storage for Dunbar-Based P2P Online Social Networks,” OTM 2014 Conferences, 2014.

[17] Renato J. Figueiredo, P. Oscar Boykin, Pierre St. Juste, David Wolinsky, “Social VPNs: Integrating Overlay and Social Networks for Seamless P2P Networking,” Advanced Computing and Information Systems Lab, University of Florida.

[18] Sean Rhea, Dennis Geels, Timothy Roscoe, and John Kubiatowicz, “Handling Churn in a DHT,” USENIX Annual Technical Conference, 2004.

[19] Arkadiusz Biernacki, “Analysis of the flooding search algorithm with OPNET,” unpubli.

[20] H.Bandara and A.Jayasumana, "Collaborative applications over peer-to-peer systems–challenges and solutions", Peer-to-Peer Networking and Applications, vol. 6, no. 3, pp. 257-276, 2012 .[Online]. Available: https://en.wikipedia.org/wiki/Peer-to-peer

[21] J.R. Douceur,P. Druschel, M. F. Kaashoek, and A. I. T. Rowstron, “The sybil attack, Peer-to-Peer systems", First International Workshop, IPTPS 2002, volume 2429 of LNCS, pages 251–256. Springer, 2002

[22] Gutachten: Prof. Neeraj Suri, Ph.D,. Gutachten: Prof. Dr. Thorsten Strufe ,” Increasing Structured P2P Protocol Resilience to Localized Attacks“ Genehmigte Dissertation von Daniel Germanus, M.Sc. aus Frankfurt am Main,2015