AnonSocialMix: Anonymous P2P File Sharing Over Social Networks

Student Name: Rajdeep Mukherjee

IIIT-D-MTech-CS-GEN-MT15051
July, 2017

Indraprastha Institute of Information Technology
New Delhi

Thesis Committee
Dr. Sambuddho Chakravarty, IIIT Delhi (Advisor)
Dr. Tanmoy Chakraborty, IIIT Delhi (Internal Examiner)
Dr. Vinay Joseph Ribeiro, IIT Delhi (External Examiner)

Submitted in partial fulfillment of the requirements
for the Degree of M.Tech. in Computer Science,
in Information Security Category

Certificate

This is to certify that the thesis titled **"AnonSocialMix: Anonymous P2P File Sharing Over Social Networks"** submitted by **Rajdeep Mukherjee** for the partial fulfillment of the requirements for the degree of *Master of Technology* in *Computer Science & Engineering* is a record of the bonafide work carried out by him under my guidance and supervision in the Security and Privacy group at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Dr.Sambuddho Chakravarty

Indraprastha Institute of Information Technology, New Delhi

Abstract

Peer-to-peer (P2P) file sharing accounts for one of the major sources of the Internet traffic. As privacy and anonymity issues continue to grow due to constant censorship and network surveillance, more and more Internet users are getting attracted towards the facilities for anonymous communication. Extensive research has been conducted over the years towards the design and development of several anonymous P2P file sharing protocols and systems. Size of the *Anonymity Set* plays a crucial role in determining the degree of anonymity being provided by such networks. However, most of the existing anonymous infrastructures create a completely new network and invite users to join in. As a result, even popular systems like *Freenet* and *GNUnet* suffer from not enough participants. Popular Online Social Networks (OSNs) like *Facebook* and *Twitter* have an existing strong network of millions of users which can provide us with a readily exploitable *abstraction* of a P2P platform for implementing an anonymous communication scheme.

In this thesis, we explore the possibility of allowing censorship-resistant P2P file sharing on top of *Facebook*, an inherently non-anonymous non-P2P architecture. We present the design of **AnonSocialMix,** an overlay network that uses Facebook as the underlying platform to enable its existing set of users to search and share files in a distributed, anonymous, peer-to-peer fashion. We use *Dropbox* as the file hosting service. Users of our proposed system no longer need to hide their identities behind a mask as the proposed cryptographic framework preserves the anonymity, privacy and confidentiality of the communications and ensures an infrastructure that is strongly resistant to *Eavesdropping*, *Traffic Analysis* and *Timing Attacks*. It is further infeasible to determine the actual source of a request or a reply as the proposed communication scheme makes it impossible to distinguish between the actual creator of a message and its forwarder. We have implemented a server-based running prototype of our proposed design in the form of a *Chrome extension* to verify our claims. We further compare its performance with the results of a controlled lab based simulation of our system and obtain encouraging results.

Keywords: Peer-to-Peer Systems, Anonymous Content Sharing, Censorship Resistance, Anonymity and Privacy, Security, Online Social Networks

Acknowledgments

I would like to express my deepest gratitude to my advisor Dr.Sambuddho Chakravarty for his guidance and support. The quality of this work would not have been as high without his well-appreciated advice. I thank Sambhav Satija and Harshvardhan Kalra for their efforts in implementing the browser extension. I would also like to thank Devashish Gosain and Swati Agarwal for their critical reviews and valuable comments from time to time.
Most importantly, I would like to thank my mother for providing me the motivation, support and encouragement during the entire duration of my thesis. Last but not the least, I would like to thank IIITD for making this happen. The infrastructure, services and environment provided to us as students are truly remarkable.

I would like to dedicate this work to my father and my maternal aunt who could not live long enough to see this day.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 What is Anonymity?

Anonymity refers to hiding one's identity. Anonymity on the Internet applies to any interaction a user has on the Internet that protects his or her identity from being shared with another user or with a third party [1]. However, there is a subtle difference between anonymity and privacy. Whereas privacy refers to hiding one's actions, anonymity allows one to see what you do without knowing who you are.

1.2 Significance of Anonymity in Digital Communication

We live in the modern era of Information and Communication Technology (ICT), where Internet has become an indispensable part of our lives as a dominant means of communication. We regularly use the Internet services for various purposes like e-Commerce, e-Banking, VoIP and e-Mails, Social Networking and Content Sharing. However, increasing amounts of *censorship* and *pervasive monitoring* of user Internet access by law-enforcement agencies and governmental institutions pose serious threat to the very fundamental rights of free speech and user privacy on the Internet. As a result, anonymous networks are rapidly gaining importance in the context of digital communication as they protect people's right to online privacy by hiding the identity of senders and receivers, or the communication linkage between a sender and a receiver, thereby reducing the possibility of getting recognized and hence victimized [11]. Anonymous networks thus play a critical role in supporting free speech and privacy on the Internet.

1.3 Background

Content sharing and information dissemination are one of the most dominant types of online communication among netizens. However, the TCP/IP protocol suite does not inherently provide any privacy for data transfer over the Internet [21]. As issues of data corruption, information leakage and privacy infringement continue to rise, more and more users are gradually realizing the importance of keeping their online activities anonymous. In this context, extensive research has been conducted over the years towards the design and implementation of Anonymous Peer-to-Peer (P2P) file sharing systems. In this dissertation, we use the terms *file* and *content* interchangeably.

Users of a P2P system communicate and exchange content directly with each other over the Internet without the need of a centralized server or authority. Such systems typically have inherent advantages of being resistant to censorship and centralized control, fault tolerant, scalable and offer increased access to resources. Protocols like *BitTorrent* achieve the objectives of easy and fast file sharing between peers, but they do not provide anonymity to its participants. Anonymous P2P content sharing systems additionally provide certain degree of anonymity to the users while enabling them to contribute, search and obtain digital content in a distributed and coordinated manner. On the basis of "degree of centralization", P2P file sharing architectures, in general, can be classified as *Purely Decentralized* (such as Freenet and original Gnutella architecture), *Partially Centralized* (systems such as Kazaa and Morpheus) and *Hybrid Decentralized* (such as Napster) [20].

Over the years, several schemes and algorithms have been proposed for the design and development of anonymous networks especially anonymous P2P file sharing systems. These include MIXes, Onion Routing, Crowds, per-hop source address rewriting and message forwarding, UDP address spoofing and various cryptographic algorithms. A no. of practical systems have been implemented and deployed based on one, or a combination of these methods. Some of the well known systems include Anonymous Peer-to-peer File-Sharing (APFS) and Tor (based on Onion Routing), the Invisible Internet Project (I2P) (based on Garlic Routing), Free Haven and GNUnet (based on MIXes), Freenet, Pisces, OneSwarm and Garlic Cast. Most of these systems are "friend-to-friend" networks and trade efficient routing for anonymity. A given node (peer/user) only has a limited view of the whole network as it connects to a small no. of other known nodes. Hop-to-hop query forwarding across such overlay networks enables communication with remote nodes [6].

1.4 Research Motivation

Peer-to-peer (P2P) file sharing accounts for one of the major sources of the Internet traffic and has received considerable research attention over the years. Furthermore, controversies over Napster and Gnutella and recent legal actions against Megaupload, a major file-sharing website facilitate a stronger demand for anonymous distributed file storage and peer-to-peer sharing services. Anonymous P2P file sharing is often linked with illegal or copyrighted downloads. As privacy and anonymity issues continue to grow, maintaining a certain degree of anonymity online has now become imperative for the netizens to avoid censorship and surveillance by powerful institutions and protect themselves from the fear of prosecution or victimization while sharing files with peers over the Internet or participating in uploading / downloading of unofficial leaks, among other online activities [8].

This very concept has motivated extensive research over the years towards the design and development of several anonymous P2P file sharing protocols and systems. Size of the *Anonymity Set* plays a crucial role in determining the degree of anonymity being provided by such networks. However, most of the existing anonymous infrastructures create a completely new network and invite users to join in. As a result, even popular systems like *Freenet* and *GNUnet* suffer from not enough participants. To the best of our knowledge, none of these systems try to exploit the strong user base provided by the popular Online Social Networks (OSNs) like *Facebook* or *Twitter*.

In this thesis, we propose the design of an overlay network that uses Facebook as the underlying platform to enable its users to search and share files in a distributed peer-to-peer fashion by posting encrypted messages on the wall of a given Facebook group. We have selected *Dropbox*, given its security features, as the file hosting service to be used by any user willing to publish his / her own files. Members of the group participating in file sharing no longer need to hide themselves behind a mask as the proposed cryptographic framework ensures the anonymity of communications even in the presence of a global or local adversaries. Given the ease of use and the security guarantees of our proposed system, we believe that more and more users will be attracted to use it simply by installing a *Google Chrome Extension*, developed as a part of this work, in their respective web browsers and using their existing Facebook credentials to log in. We hope that this work enables more and more netizens with good intent to enjoy the benefits of anonymous P2P file sharing over the Internet without any fear of censorship and victimization.

1.5 Organization of the Thesis

The remainder of the thesis is organized as follows. In Chapter 2 we give an overview of anonymous networks and highlight their significance in the modern era of digital communication. We then briefly discuss about anonymous peer-to-peer (P2P) communication networks and pay particular attention to anonymous P2P content sharing networks (being the principal focus area of research in this work). Few popular protocols and implemented systems in both the categories are described. Chapter 3 defines the problem statement and highlights our contributions. Design and implementation details are described in Chapter 4. It further outlines the threat model and describes the advantages of our proposed system. Chapter 5 describes the experimental setup and compares the results of the simulations carried out in an ideal and controlled lab environment with those of the actual running prototype. Chapter 6 discusses the limitations of our proposed design and outlines the future work. Chapter 7 concludes this dissertation.

Chapter 2

Related Work

2.1 A Brief Survey of Anonymous Networks

Anonymous networks enable users to access the Web or communicate over the Internet by protecting their identities from getting traced, thereby reducing the possibility of getting recognized and hence victimized. Such networks make network surveillance and traffic analysis more difficult, if not completely prevent them. Anonymous networks thus play a crucial role in the context of digital communication by promoting and protecting people's right to online privacy and freedom of expression.

- **Why are Anonymous Networks Important?**
  "Man is least himself when he talks in his own person. Give him a mask, and he will tell you the truth". This quote by Oscar Wilde literally sums up the importance of maintaining a certain degree of anonymity while taking part in any form of online activity that can be monitored or tracked and hence can pose a serious threat to our very fundamental rights of free speech and user privacy on the Internet. *Investigative Journalism*, *Whistleblowing* on colleagues or superiors for reporting illegal activities, *Anonymous Peer Review*, helping out the Police in *Law Enforcement* by providing information about criminal activities, drug sales and related activities, calling for *Self Help* from peers regarding matters such as family or sexual abuse, suicidal thoughts due to alcoholism or drug abuse, mental and physical illness etc. and *Avoiding Persecution* at the hands of repressive regimes are some cases where anonymous networks can play a huge role by allowing users to express themselves freely without the fear of retaliation or getting judged [17].

Most of the anonymous networks are distributed, peer-to-peer (P2P) systems in which nodes or participants are anonymous or pseudonymous. Users of a P2P system communicate and share content directly with each other over the Internet without the need of a centralized server or authority. Anonymity of participants is achieved by special routing overlay networks that hide the identity of each node from its peers. Communication with remote nodes is done by sending messages hop-by-hop across such an overlay network [6]. The principle motivation behind using such a network is aided by the anonymous or pseudonymous nature of the nodes which makes it difficult, if not impossible, to determine whether a node sending a message is the actual originator of the message or is simply forwarding it on behalf of some other node. In order to maintain this core concept of *plausible deniability* and anonymity, it is imperative more every node in an anonymous P2P network to act both as a universal sender as well as a universal receiver [2].

- **Opennet vs Darknet Modes of Operation**

  Based on the way in which a node selects its peers, an anonymous P2P system can operate in two different modes namely *Opennet* and *Darknet* (often referred to as Friend-to-Friend). In **Opennet**, there is very little or no control on peer selection. A node gets connected to randomly discovered peers and immediately becomes visible to the entire network which might even include compromised nodes. Whereas, in **Darknet**, a node only gets connected to its friends whom it trusts. Darknet usually requires more efforts to set up but is safer than Opennet as a node is only visible to its trusted peers and invisible to the rest of the network. **Freenet** supports both the modes.

  Our proposed system considers a trust based framework where trust or reputation of a node is calculated on the basis of scores given to it depending upon its quality of responses. However a new node joining the network for the first time randomly selects peers to start communication. Scores and associated trust levels are gradually updated over time.

- **Communication vs Content Sharing Systems**

  Depending on the kind of application a network is used for, anonymous P2P networks can be classified as Anonymous Communication Systems and Anonymous Content Sharing Systems. Whereas, the former usually support low latency, real time applications like Web browsing, the later support file sharing applications that can tolerate a certain amount of network delay. Following sections discuss these networks in greater detail.

2.2 Anonymous P2P Communication Systems

Anonymous P2P Communication Systems provide a secure bridge between a content retriever and a content provider. These are low latency, responsive networks supporting real time applications like Web browsing. What differentiates such networks from anonymous P2P content sharing systems is here the retriever knows exactly where the content is stored. However, instead of sending the query directly to the content server, the message is routed through a chain of proxies. Each proxy is only aware of its immediate neighbors. Hence, only the first proxy along the path knows the initiator of the query and the last proxy knows the real destination of the query. We now discuss Onion Routing which is an infrastructure for private communication over a public network. We then discuss about Tor, which is a a circuit-based low-latency anonymous communication service built on top of Onion Routing protocol.

2.2.1 Onion Routing with Tor

Onion Routing was developed in the mid-1990s as an infrastructure to facilitate bi-directional, low latency anonymous connections (over public networks) that are strongly resistant to both eavesdropping and traffic analysis [14]. Here, the initiator of a query knows about its destination. However, connection with the responder is established through a random sequence of machines called *onion routers*. The route, though is strictly defined at connection setup by means of constructing a recursively layered data structure called *onion*.

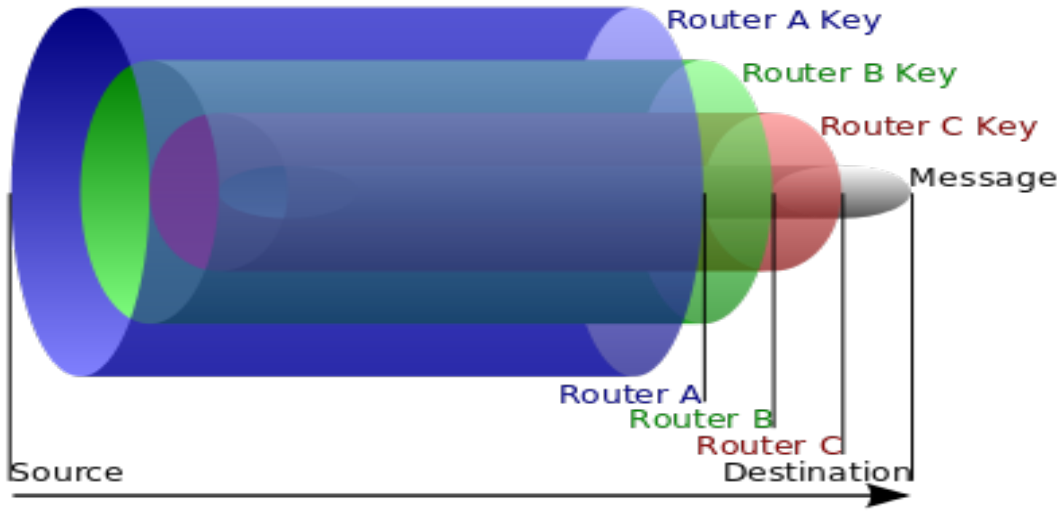


Figure 2.1: Onion Data Structure

An onion is formed by wrapping the message with successive layers of encryption. Each layer is encrypted with the public key of the corresponding onion router. An onion router, upon receiving an onion, peels off the outermost layer of encryption with its private key which reveals the identity of the next router along the path. Owing to multiple layers of encryption, each with a different key, data appears different to each onion router as it moves through the network. An intermediate router thus obtains no additional information about the path apart from the identity of its predecessor and successor. The initiator of the query thus remains anonymous to the responder since no intermediary knows both the origin and final destination of the data.

Tor stands for **T**he **O**nion **R**outer, a circuit-based low-latency anonymous communication service [18] originally developed by the U.S. Naval Research to protect governmental communications. Tor is a volunteer-operated anonymity network run by a diverse set of organizations and individuals donating their bandwidth, processing power and technical expertise. The project is maintained by *The Free Haven Project*, and its web resources are donated by the *Electronic Frontier Foundation*. The main objective of this project is to develop a distributed communication network *overlaid on the Internet* which conceals a user's location and usage from anyone conducting network surveillance or traffic analysis by directing the Internet traffic through a free, worldwide overlay network consisting of more than seven thousand relays [5].
Tor supports user anonymity through a chain of proxies or relays. A user needs to create a virtual circuit before it actually sends a request message to a remote server. In Tor, every communication channel contains exactly three relays: an entry, an intermediary, and an exit.

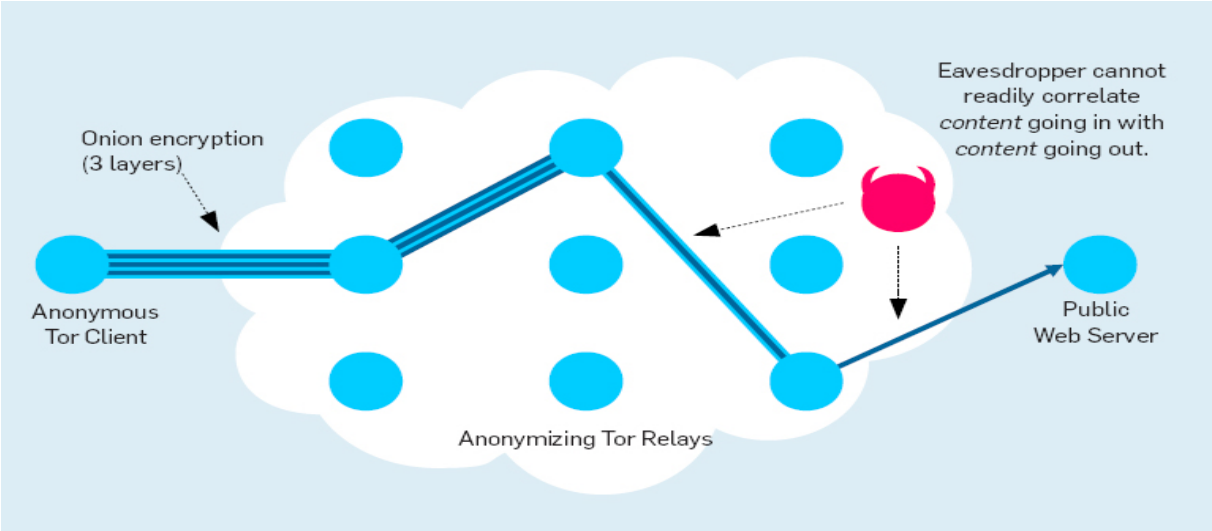


Figure 2.2: 3 relays in Tor

To begin with, Alice's Tor client obtains a list of Tor nodes from a directory server. It then picks a random path to the destination server. Tor software incrementally builds a circuit of encrypted connections through relays on the network. The circuit is extended one hop at a time, and each relay along the way only knows which node gave it data and which server it is giving data to. No individual relay ever knows the complete path that a data packet has taken. Tor software on user's machine negotiates a separate set of encryption keys for each hop along the circuit to ensure that each hop can not trace these connections as they pass through [4].

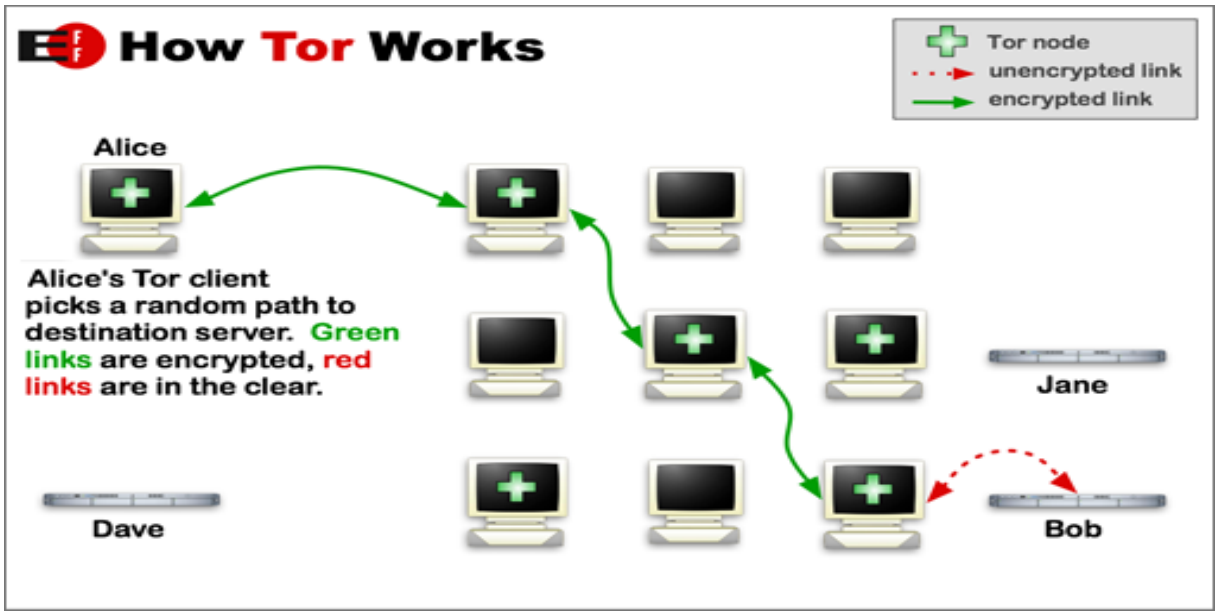


Figure 2.3: How Tor Works

2.3 Anonymous P2P Content Sharing Systems

Peer-to-peer (P2P) distributed computer architectures allow users to communicate and exchange resources (storage, content, CPU cycles etc.) directly with each other over the Internet without the need of a centralized server or authority. Such systems typically have inherent advantages of being resistant to censorship and centralized control, fault tolerant, scalable and offer increased access to resources even in the presence of a highly transient population of nodes, network, and computer failures.

P2P file sharing and information dissemination account for one of the major sources of the Internet traffic and have received considerable research attention. Most of the P2P systems are content sharing applications which enable their users to securely and efficiently publish, organize,

index, search, update, and retrieve digital content in a distributed and coordinated manner. On the basis of "degree of centralization", P2P file sharing architectures, in general, can be classified as *Purely Decentralized* (such as Freenet, the KAD network and the original Gnutella architecture), *Partially Centralized* (systems such as Kazaa and Morpheus) and *Hybrid Decentralized* (such as Napster).

As issues of censorship, pervasive monitoring, data corruption, information leakage and privacy infringement continue to rise, it has become extremely crucial for the netizens to keep their online activities anonymous in order to protect their very own rights to express and communicate freely online without the fear of retaliation or getting unjustly victimized or judged. More and more users are gradually realizing the need to hide their real identities while sharing confidential content online with their peers related to private or sensitive issues. In this context, extensive research has been conducted over the years towards the design and implementation of Anonymous Peer-to-Peer (P2P) content sharing systems. Majority of such systems are high latency, delay tolerant *friend-to-friend* networks where each node (or peer) connects with only a small no. of other known or trusted nodes. Except for its immediate neighbors, a node's visibility is hidden from the rest of the network. A content retriever does not know where the desired content is. Communication with remote nodes is accomplished by sending messages hop-to-hop across such an overlay network.

Over the years, several schemes and algorithms have been proposed for the design and development of anonymous networks especially anonymous P2P file sharing systems. Such overlay networks usually define their own network topology and protocol for inserting and searching contents. A no. of practical systems have been implemented and deployed based on one, or a combination of these methods. We discuss some of the relevant protocols and systems in the upcoming sub sections.

2.3.1 Crowds

Crowds is an anonymous protocol for web-transactions proposed by Reiter and Rubin [16]. This protocol involves a group of users, called a "crowd", each of whom wants to communicate with a corresponding web server but without revealing his identity. The idea is to randomly route each message through the crowd until one member of the crowd decides to pass it to the server. This ensures that neither the receiver nor the nodes in the system can tell who sent the message. This system requires all nodes to be connected to all other nodes, and so it scales badly to larger

networks.

2.3.2 GNUNet

GNUnet is a peer-to-peer content sharing network. Its design goal is to provide strong user anonymity and censorship-resistance. In GNUnet, each node contributes a portion of its hard disk to the global network storage. Nodes can join and depart from GNUnet dynamically at any time without approval by a certified authority or central server.

In terms of routing, GNUnet uses random routing along with Kademlia routing. Slightly different from Kademlia, GNUnet's message routing is actually carried out in two stages. In the first stage, a request message is routed randomly in the network. After traversing a sufficient number of hops (roughly log(n), where n is the number of nodes in a GNUnet network), in the second stage, the request message is forwarded according to the Kademlia protocol, with an exception that, the routing is carried out in a recursive fashion instead of an iterative fashion as in the original Kademlia system, due to the anonymity requirement of GNUnet. Random routing in the first stage helps to improve the overall anonymity strength of GNUnet. GNUnet utilizes a credit-based economic system for message routing. The goal is to prevent denial of service attack by limiting the resources available to an attacker. In addition, GNUnet employs a special mechanism, known as *shortcut*, to maintain load balancing and reduce network latency. After a node receives a query message from a neighbor node, it chooses to either indirect or forward the query if it does not possess the query results. *Indirect* a message means overwriting the return IP address with its own address. *Forwarding* does not change the return address of the upstream node. Figure 2.4 illustrates the concepts of indirecting and forwarding.

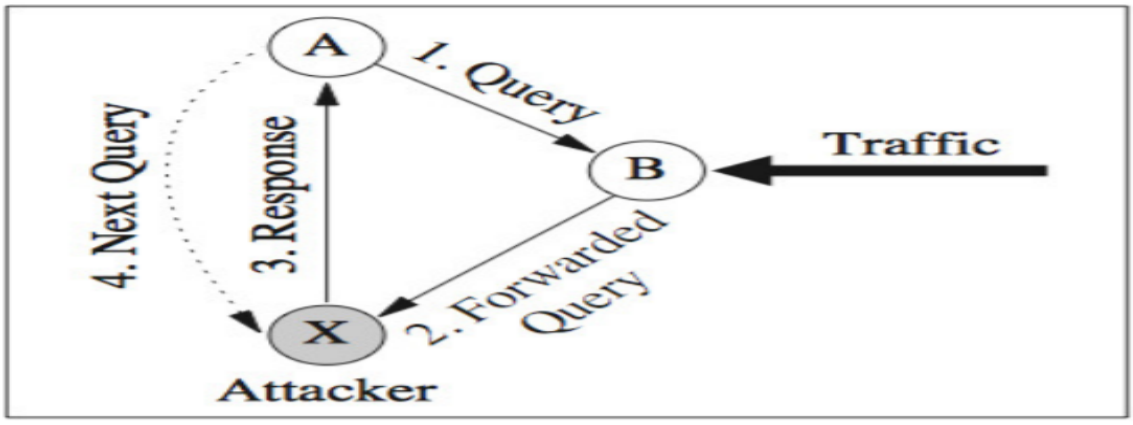


Figure 2.4: Indirecting and forwarding

2.3.3 Freenet

Freenet [12] is a search-able peer-to-peer system for censorship resistant document storage. It is both an original design for anonymity and an implemented system. While it does not aim to hide the provider of a particular file it does aim to make it impossible for an attacker to find all copies of a particular file. A key feature of the Freenet system is that each node will store all the files that pass across it, deleting the least used if necessary. A hash of the title (and other key words) identifies the files. Each node maintains a list of the hashes corresponding to the files on immediately surrounding nodes. A search is carried out by first hashing the title of the file being searched for, and then forwarding the request to the neighboring node that has the file with the most similar hash value. The node receiving the request forwards it in the same way. If a file is found, it is sent back along the path of the request. This unusual search method implements a node-to-node broadcast search one step at a time. Over time it will group files with similar title hash values, making the search more efficient.

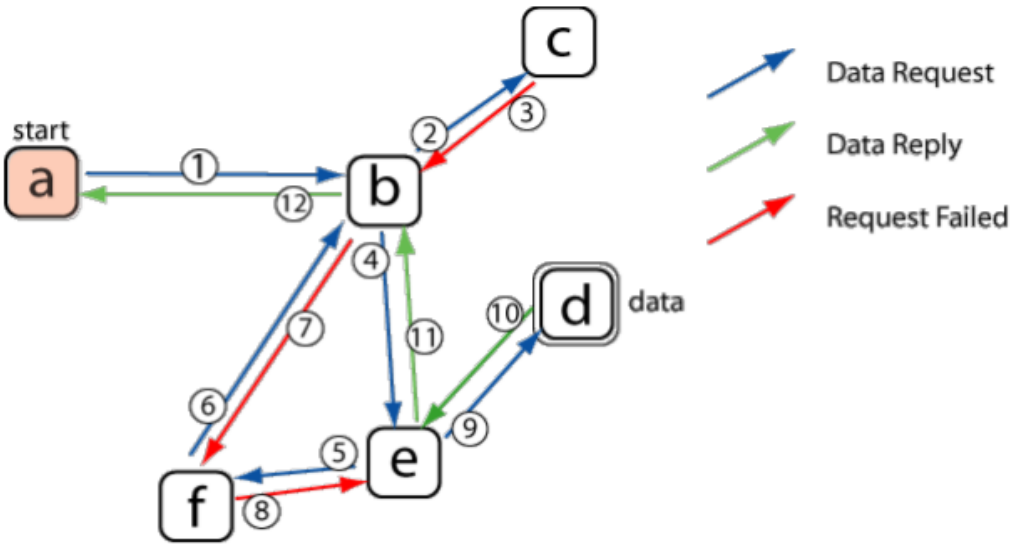


Figure 2.5: Freenet Routing Scheme

Chapter 3

Problem Statement and Contribution

3.1 Problem Statement and Design Goals

Peer-to-peer file sharing has been one of the most dominant sources of Internet traffic over the last two decades. Extensive research has been carried out towards the design and development of several anonymous P2P file sharing protocols and systems that offer some kind of anonymity to their users. Most of these systems create a new network and expect users to join in. Users on the other hand might hesitate to trust and join a completely new network just for the sake of sharing files anonymously with peers. Our goal in this work is therefore to come up with a novel design of a censorship-resistant P2P file sharing protocol that meets the following criteria:

- It must provide the users of an existing network, that is not inherently P2P, an abstraction of a large P2P network by enabling them to search and share files in a distributed peer-to-peer fashion. Size of the anonymity set is thus maintained.
- It must be resistant to *Traffic Analysis*, i.e. though the sender and receiver can each be identified as participating in some communication, they cannot be identified as communicating with each other.
- It must ensure *Probable Innocence* whereby from the attacker's point of view, the sender appears no more likely to be the originator of the message than to not be the originator.
- The design must be deployable and usable in the real world.

3.2 Contribution

In this work, we explore the possibility of allowing censorship-resistant P2P file sharing on top of an inherently non-anonymous non-P2P platform like Facebook thereby trying to exploit its strong user base to give us a strong enough *Anonymity Set* and in the process making our system more *Usable* [18]. We propose the design of **AnonSocialMix:** an overlay network that uses Facebook as the underlying platform to enable its existing set of users to search and share files in a distributed, anonymous, peer-to-peer fashion where anonymity in our case refers to *Unlinkability of sender and receiver* [16]. The salient features of our system are as follows:

- We have selected Dropbox, given its security features, as the file hosting service to be used by any user willing to publish his / her own files.
- Members of the given Facebook group participating in file sharing no longer need to hide themselves behind a mask or a pseudonym as the proposed cryptographic framework makes it impossible for a *Global Adversary* to decrypt and infer the nature (request / response) of a random post. It further ensures the *Unlinkability of sender and receiver* criterion thereby making the system resistant to Traffic Analysis attacks.
- Even in the presence of a *Local Adversary*, the criterion of *Probable Innocence* is ensured as there is no way to distinguish between the actual creator of a message and its forwarder.
- We propose a *Reputation* based system whereby a user tags only its most reputed / trusted peers while posting an encrypted search query on the group wall. Reputation of a node is gradually updated with positive or negative scores based on the quality of responses it provides. Hence, a malicious user which only searches the network and doesn't respond to queries won't survive for long as its reputation score will eventually fall off and its peers might decide not to further respond to its queries.
- We have implemented a server-based running prototype of the system in the form of a *Google Chrome extension* to verify our claims. The prototype is already deployed on one of our servers at IIIT Delhi. We further compare its performance with another version of simulation code written to test the working of our proposed design in an ideal controlled lab environment and obtain encouraging results in the process.

Chapter 4

AnonSocialMix

In this section we present the detailed design of AnonSocialMix. We start with a general overview of the proposed overlay network followed by a description of the threat model considered. We then present the design and implementation details of our proposed architecture.

4.1 Problem Overview and Assumptions

AnonSocialMix presents an overlay network which enables existing Facebook users to anonymously share content in a decentralized peer-to-peer fashion. A user performs as a node in such a framework. We define *initiator* as the node which is looking for a content and *provider* as the node which provides a content being searched. All the interested users join a given Facebook group by logging in to our system using their existing Facebook credentials. An initiator searches the network for its desired content by means of posting an encrypted query message on the group wall after tagging its most reputed peers. The message gets re-posted by the intermediary nodes till it finds a content provider which then replies back with the appropriate content. Though the real identities of the nodes are not hidden, however the communications remain anonymous where we define anonymity as initiator-responder unlinkability.

We assume that users of our system must have a Facebook account for querying, and a Dropbox account for replying back to a file request, as the search responses in our case are encrypted download-able Dropbox file links. A free riding user, interested in only searching and not participating in relaying queries our replying back to queries, would only require a Facebook account to use our system. However, such nodes won't be able to survive for long as the system takes into account the reputation score of a user which is earned by replying back to search queries.

4.2 Threat Model

As against related works like [18] [10] and [19], we consider a global passive adversary which can observe all the communications taking place in the concerned Facebook group. It's main goal is to link the responder of a message with its initiator and, upon confirming the suspicion, victimize the participants. A message, before getting posted on the wall, is encrypted with a key that is shared only between the two immediate neighbors participating in the current hop. With each successive hop / re-posting, the message gets encrypted with a different shared key and hence looks completely different. It thus becomes impossible for such a global adversary to decrypt and infer the nature (request / response) of a random post and further correlate the posts to trace the entire communication path between the initiator and the responder. This makes our system resistant to *Traffic Analysis* attacks.

Further, A Facebook user with malicious intent can always join the group as a local adversary. A passive attacker can decrypt the network traffic being forwarded through it and hence try to find a link between the initiator and the responder. However the maximum information it can decipher, by decrypting a post meant for it, is the nature of the post and who sent it. This information is however insufficient to determine whether the sender is the actual creator of the message or simply forwarding it on someone else's behalf. *Probable Innocence* is, thus still ensured and the communication scheme remains anonymous as there is no way to distinguish between the actual creator of a message and its forwarder.

An active local adversary, can on the other hand generate, modify, delete or delay traffic. It can further collaborate with other active malicious nodes in order to mount *Statistical Attacks* or *Timing Attacks*. However, our system ensures that noise is added with each message to obfuscate the posts. Also, random noisy posts are made on the group wall from time to time by random nodes to make it harder for the adversaries to correlate the network traffic thereby further reducing the possibility of such attacks.

Our proposed system is vulnerable to *Distributed Denial of Service (DDoS) attacks* where a collection of malicious nodes collaborate to flood the network with random query posts thereby forcing the legitimate users to wait for longer durations of time to get a response to their queries or none at all. Again, similar to most overlay anonymous communication protocols [15] [19] [10], AnonSocialMix is vulnerable to *Sybil attacks* [7] as the fraction of malicious nodes can arbitrarily attain a value close to 1 in the presence of Sybil attackers. Protocols like *SybilGuard* [9] can be used to further improve the robustness of the system.

4.3 System Design

4.3.1 Description of Functions

- H (text) represents the Hash of the "text". We have used MD5 hashing technique in our implementation.
- HMAC (data) represents a message authentication code (MAC) based on a hash function where MAC is produced from the "data" and a secret key, shared secret between the two communicating parties in this case. HMAC serves as a signature which is used to authenticate the integrity of the data shared between the two parties.
- Enc (data, key) represents the encryption of "data" under "key". In our implementation, we have used AES-256 symmetric-key encryption algorithm to encrypt the network traffic.

4.3.2 Bootstrapping

A new user upon joining the group, selects a random set of five or less existing group members to communicate with and initiates a key negotiation request individually with each of them by taking part in a 3-way handshaking mechanism similar to TLS / SSL V3. Encrypted ***keyneg*** request messages are tagged with the intended recipients and posted on the group wall.

4.3.3 Establishing Shared Secrets between Two Peers

- Let *Alice* initiate a key negotiation request with *Bob*. Alice creates a random text R_a, encrypts it with Bob's *OpenPGP* public key Pu_b and posts the encrypted message on the group wall after tagging Bob with it.
- Bob, upon receiving the notification scrapes the wall and receives the *keyneg* request from Alice. Bob, in turn creates a random text R_b, encrypts it with Alice's OpenPGP public key Pu_a and posts the encrypted message on the wall after tagging Alice with it.
- Alice receives R_b. She then creates a random key **"S"** to be shared with Bob.
- Alice calculates K_{ab} = H ("S", R_a, R_b, "CLNT"). She further calculates X_{ab} = HMAC (R_a, R_b) | K_{ab}, where "|" represents text concatenation operator.
- Alice sends Enc ("S" | X_{ab}, Pu_b) to Bob.

- Bob decrypts the post with its private key Pr_b and obtains "S". It further calculates X_{ab} in a similar fashion as Alice did and compares it with the received X_{ab} in order to verify the integrity of the message received. This completes key negotiation between the two parties. Alice and Bob encrypt further messages between them using the shared secret "S".

- The key "S" shared between Alice and Bob is not permanent. Both the communicating parties save the time stamp of the last key negotiation between them. New shared key is exchanged between the two as soon as the previous one expires after a given duration of time, half an hour in our implementation.

4.3.4 Score Vectors and Reputation Calculation

Every member of the group has a list of trusted peers which is empty when the user joins the group and eventually grows in size as communications take place over time. Each peer in the list is associated with the following attributes:

User Name	Reputation Score	Last Communication Time	Shared Key	Last Key Negotiation Time

Figure 4.1: Peer List

Score Vector consists of the first two columns i.e. *User Name* and *Reputation Score* of the above matrix. Score Vectors are shared whenever two users negotiate a key between them. For e.g. if Alice and Bob were exchanging keys, then sharing of score vectors by both the entities enables Alice to discover Bob's trusted peers and vice versa. This helps both of them to further discover the network and gives them more options of peers to select from when searching the network for a file.

Reputation Score of a peer is calculated by giving equal weightage to the group perception and one's own perception about that peer based on the quality of responses it provides. For e.g. let both Alice and Bob have a common peer *Charlie* in their respective *Peer Lists*. Let the score given by Alice to Charlie be $\mathbf{x}$ and the same given by Bob to Charlie be $\mathbf{y}$. When, score vectors are exchanged between Alice and Bob, then both the entities update Charlie's reputation score to $\mathbf{(x + y) / 2}$ in their respective peer lists.

Scores further need to be *normalized* such that the reputation score of a random user doesn't blow out of proportion. This will avoid a malicious user to initially play good and gain a very high reputation score before suddenly going down and affecting the communications of the rest of the group at least for some duration of time. We have currently kept the scores in the range

of -10 to +10. As the scores will be manually entered by the users through the Google Chrome extension, hence we have ensured that even if a user enters an abruptly high or low score, we immediately bring it within the pre-decided range.

4.3.5 File Creation and Upload

A user willing to publish her own files first uploads them on her own Dropbox account and obtains their shareable links. Each file must be associated with a set of keywords with which it can be searched. Next she uses our system to enter the shareable file URLs along with their associated keywords. Following structure of database is maintained internally.

File URL	Keywords (k1, k2, .., kn)	Hash of Keywords	Double Hash of Keywords

Figure 4.2: Database Structure

4.3.6 Selecting Peers for Tagging

Maximum of 5 peers can be selected at a time for tagging a post (search query / response) owing to a limitation on the length of a post that can be posted on any given Facebook wall. If n users are to be selected, then top n-1 users are selected on the basis of reputation scores. The last one is chosen as the one having the oldest communication time. This is done to ensure that the same set of users does not get selected every time. This further provides an opportunity to a peer which might have faired poorly in the past but now has the potential to perform better.

4.3.7 Searching for a Content

- Alice selects a set of keywords k1, k2, k3...,kn to search for a file. She enters a comma separated list of these keywords in the browser extension and hits the "Search" button.
- The extension internally creates the search query message
  M = H (H (k1)) | H (H (k2)) | H (H (k3)) | ... | H (H (kn)). The searches are thus made with double hashes of keywords. The significance of this step will be better understood when we discuss the response mechanism.
- Alice now selects top 5 (or less) reputed peers, as discussed above, from its peer list with whom she already has shared secrets available.

- M is encrypted individually with each of the shared secrets. Noise is added to further obfuscate the message. The entire encrypted message Q is tagged with the above selected peers.

- *Steganography techniques* are applied to convert this gibberish looking encrypted text into English looking words before posting the query message on the group wall.

- Every request message is associated with a *unique ID*. The responder while replying also associates the response message with the same ID. Any user receiving the same request twice or more through different paths will simply ignore the messages silently.

- Every request message is further associated with a random value emulating the concept of TTL. This value is reduced by 1 every time the message is re-posted. The message is stopped from getting further re-posted as soon as this value becomes 0. This will prevent the same request from getting re-posted on the group wall infinitely.

4.3.8 Responding to a Query

Users get notified as they are tagged with posts on the group wall. An FCFS queue is maintained by each user to store the messages with which she was tagged. Requests are responded back in the same order in which they were originally received. Upon retrieving and decrypting a message meant for her, Alice might face one of the following scenarios:

- **Scenario 1: Key Negotiation Request**
  In this case, Alice shares her score vector with the requester and takes part in key negotiation as already discussed in previous sections.

- **Scenario 2: Key Negotiation Response**
  In this case, Alice completes the key negotiation process by setting the shared key with the responder and updating its own peer list with the one received.

- **Scenario 3: Search Query, no matching results found**
  In this case the query is simply forwarded ahead by re-posting it on the wall in a fashion similar to posting a search query.

- **Scenario 4: Search Response to someone else's request**
  In this case the response is simply cached against the double hashes of keywords with which

the search was made. However, Alice has no mechanism to decrypt the cached response as the search responses are encrypted with single hash of the keywords and obtaining single hashes from their corresponding double hashes is infeasible, Hashing being a *one-way* function. The contents of the response thus remain hidden from an unauthorized intermediary. Alice, further forwards the response to the node from which it had originally received the request.

- **Scenario 5: Search Query, matching results found, however not the owner of the files.**

  In this case Alice simply responds back with the encrypted contents of the match, cached previously while forwarding someone else's reply, to the node from which it received the request. Alice, however has no idea about the actual contents of the response that it sends.

- **Scenario 6: Search Query, matching results found, owner of the files.**

  There can be multiple files matching with at least one of the keywords with which the search was made. Let the search be originally made with three keywords k1, k2 and k3. Let F1 match with k1 and k2 and F2 match with k3 where F1 and F2 are shareable Dropbox file links owned by Alice. It is important to note here that Alice could obtain the plain text keywords from their double hashes because Alice's database already contained the required mappings. Alice forms the reply message as R = Enc ((k1: F1, HMAC(F1)), H(k1)); Enc((k2: F1, HMAC(F1)), H(k2)); Enc((k3:F2, HMAC(F2)), H(k3)).

  Hence, we observe that the response messages are encrypted with the single hash of the matching keywords. This ensures that only the actual initiator of the search query and the responder know the actual contents of the message. No intermediary can decipher the single hashes of the keywords from their corresponding double hashes using which the search was made. Let Alice receive the immediate request from Bob. The reply message R is encrypted with their mutually shared secret "S" and the encrypted response is posted on the group wall after tagging Bob with it.

- **Scenario 7: Search Response to my request.**

  Let Alice receive the response from Bob. A list of all the file links sent by Bob are displayed on the browser extension at Alice's end. Alice goes through all the links and individually scores them depending upon their quality and relevance to the search. Finally, Alice updates Bob's reputation score with the average of all the scores.

4.4 Advantages

We believe that the following features highlight the advantages of our proposed system:

- Unlike Freenet and GNUnet, AnonSocialMix does not suffer from lack of sufficient users as our novel approach to censorship-resistant anonymous P2P file sharing exploits a readily available strong user base of Facebook, an existing popular online social network.
- Users do not need to join a completely new network for participating in anonymous P2P file sharing. Neither do they need to hide their identities behind a mask as the anonymity of their communications is ensured by the underlying cryptographic framework.
- The running prototype of AnonSocialMix developed in the form of a Google Chrome extension has a very easy to use interface. Users can simply install the browser extension, use their existing Facebook credentials to log in to our system and start collaborating without any fear of censorship or victimization.

Chapter 5

Simulation Results and Discussion

5.1 Methodology and Prototype Implementation

We have implemented two separate versions of code in order to evaluate our system and verify our claims. The controlled lab version is implemented in Java whereby the machine runs multiple threads to emulate multiple nodes in an overlay. We incrementally evaluate this version with 50, 100, 150 and 200 users and obtain the results. We have also implemented a server based prototype system of AnonSocialMix written in Python. The prototype is already deployed on one of our servers at IIIT Delhi. We test this prototype with 50 real Facebook users and compare the corresponding results with those of the controlled lab version for the same no. of users.
To maintain consistency across the two versions, we have considered a set of total 100 files and 100 keywords randomly distributed among the users. Each user is assigned a random set of 5 files to begin with where each file is randomly associated with a minimum of 1 and maximum of 5 keywords.

5.2 Performance Criteria

- **Average Network Discovery Time**

  We say that Alice has discovered Bob and vice versa when they successfully negotiate keys for the very first time. We try to calculate the time it takes for each node in the group to discover a predefined fraction of the total population of nodes, N. We run our simulation till every node in the group has discovered at least the predefined fraction of nodes. We run our simulations for the following fractions of the network: N/32, N/16, N/8, N/4 and

N/2. The importance of this metric lies in the fact that any participating node will quickly try to discover as many reputed peers as possible such that in the event of a highly reputed peer suddenly going down, a user will have other peer options to continue communication.

- **Average Response Time**

  We try to calculate the average time it takes for a query to get replied. Here again we continue running our simulations till a predefined fraction of the nodes are discovered by each member of the group and obtain the average response times in each of those cases. Under the assumption that all the users are online and any key being searched for exists with at least one of the group members, this metric will ascertain the *usability* of our system.

5.3 Controlled Lab Simulation Results

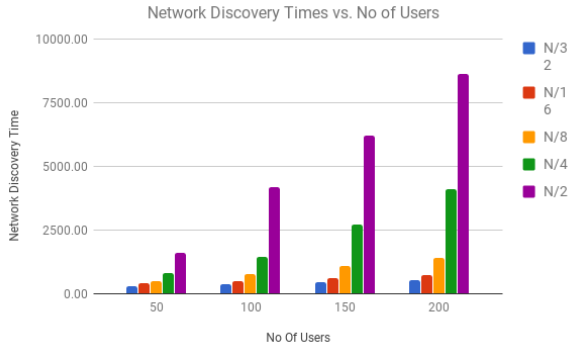
5.3.1 Average Network Discovery Times vs. No of Users

We observe that for a given no. of users, time required to discover the network grows exponentially as the size of the fraction of the population to be discovered increases from N/32 to N/2. The results are as per the logical expectation.

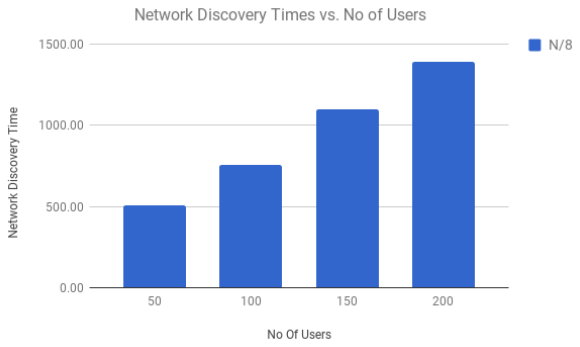
No Of Users	N/32	N/16	N/8	N/4	N/2
50	312.57	404.54	507.32	796.66	1599.64
100	361.03	481.96	758.49	1461.13	4184.85
150	446	601.14	1098.02	2702.58	6224.63
200	529.36	721.55	1393.07	4117.13	8651.56

Table 5.1: Network Discovery Times (in seconds).

Further, for a given fraction of population to be discovered, say N/8 as shown in graph 5.1 (b), network discovery times increase with increasing no. of users. The results are again logically sound as it should have taken more time to discover 2x nodes than to discover x nodes. The plot is almost linear which again confirms to our expectations that the time required to discover the network should increase linearly with linear increase in the no. of nodes in the network.



(a) Network Discovery Times: All cases



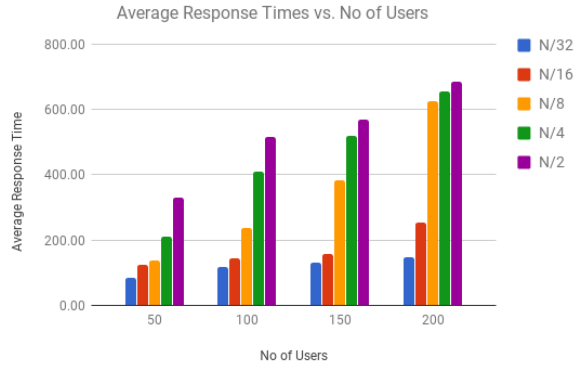
(b) Network Discovery Times: N/8

Figure 5.1: Average Network Discovery Times (in seconds) vs. No. of Users

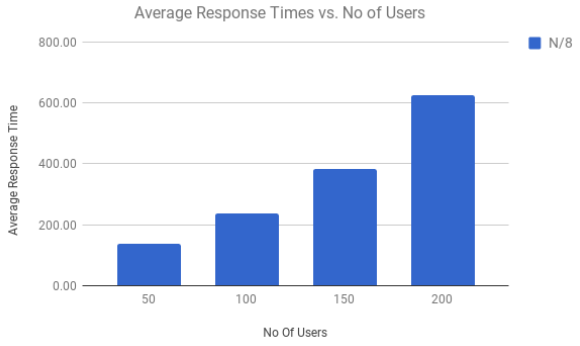
5.3.2 Average Response Times vs. No. of Users

No Of Users	N/32	N/16	N/8	N/4	N/2
50	84.71	125.58	138.8	211.98	330.53
100	116.84	144.65	236.13	410.01	515.63
150	131.33	156.21	384.5	520.91	568.05
200	147.66	253.21	626.42	656.86	686.23

Table 5.2: Average Response Times (in seconds).



(a) Average Response Times: All cases



(b) Average Response Times: N/8

Figure 5.2: Average Response Times (in seconds) vs. No. of Users

We observe that for a given fraction of population to be discovered, say N/8 as shown in graph 5.2 (b), the average response time increases with increasing no. of users. Here, one might think that as the same set of files and keywords are being distributed randomly among the users each time, availability of a resource should ideally increase with growing no of users in the network thereby reducing the average response time for a search query.

However, as the no. of users in the network increases, more and more queries get posted on the group wall. We must note here that an FCFS queue is maintained by each user to address the requests. Hence, the queuing delay for the requests made by a specific user increases in the process thereby affecting the overall response time of the queries.

This intuition is supported by the results of the following simulation which calculates the queuing delays for the requests made by a specific user, say u1, as the no. of users in the network increases. We observe that the queuing delays increase exponentially which explains the exponential nature of the graphs in fig. 5.2.

No Of Users	Average Response Time (in seconds)	Queuing Delay for u1 Requests (in seconds)
50	131.92	12.91
100	247.72	76.81
150	541.11	139.96
200	612.6	145.51

Table 5.3: Average Response Times and Queuing Delays (in seconds)

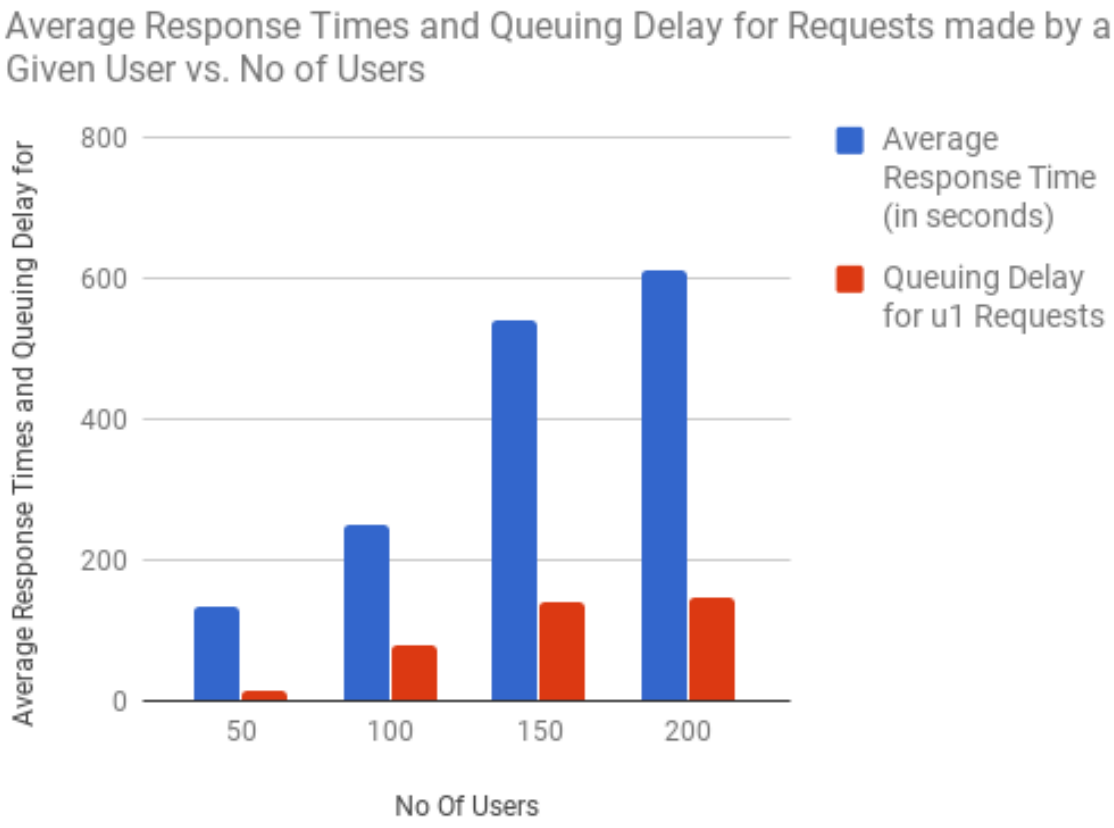


Figure 5.3: Average Response Times and Queuing Delays (in seconds) vs. No. of Users

5.4 Prototype Evaluation Results and Comparison

5.4.1 Plain text vs. Encrypted queries

We have previously discussed that our overlay network trades off efficient routing for anonymity. In this simulation, we try to compare the overhead of using our system with the average time it takes for a response while searching the network using broadcasted plain text query messages. We carry out the simulation with 50 users over a duration of 1 hour and obtain expected results. We further find that as the size of the fraction of the population to be discovered increases from N/32 to N/2, there is very little increase in the response times with plain text queries. However, the increase in the response times are significant while using our system due to the overhead of encryption, decryption and queuing delays at each hop.

Mode	N/32	N/16	N/8	N/4	N/2
Encrypted Hop By Hop	84.71	125.58	138.8	212	330.53
Plain text Broadcast	26.53	34.56	42.58	46.72	54.23

Table 5.4: Average Response Times (in seconds) vs. Mode

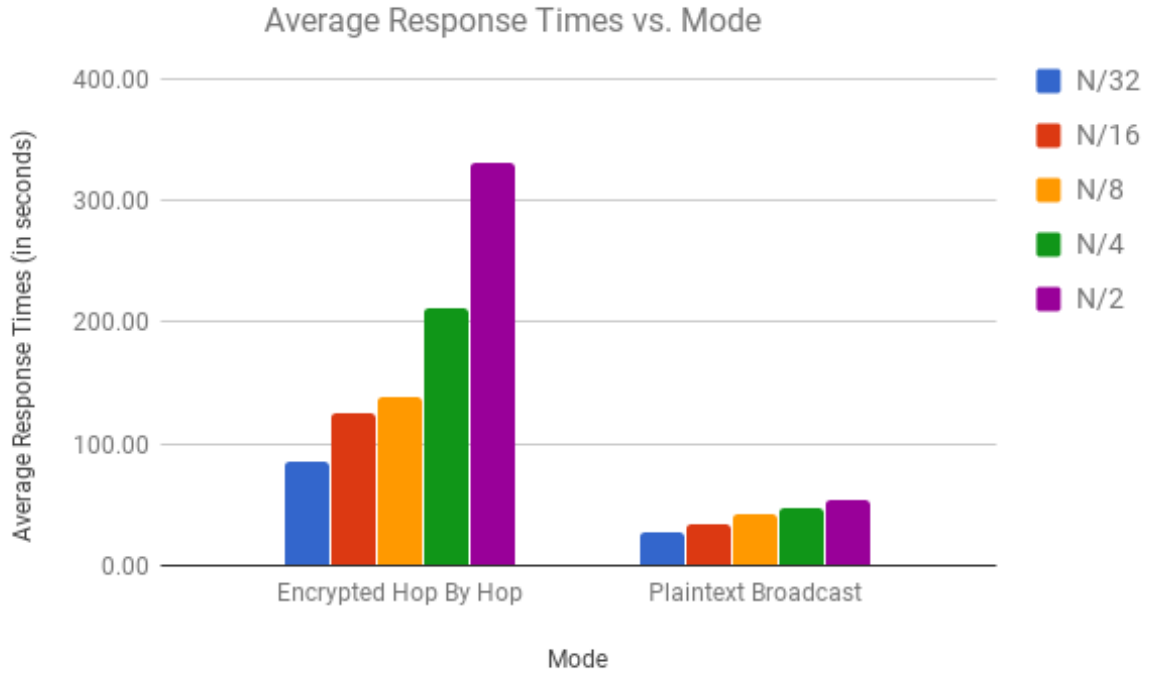


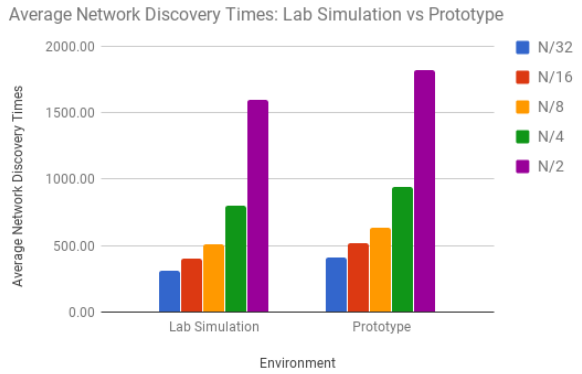
Figure 5.4: Average Response Times vs. Mode

We consider a total of 50 real Facebook users and compare the performance of our prototype with the controlled lab based simulation results with an equivalent no. of test users.

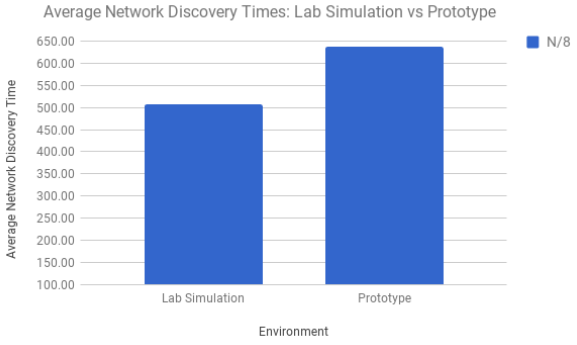
5.4.2 Average Network Discovery Times: Lab Simulation vs Prototype

Environment	N/32	N/16	N/8	N/4	N/2
Lab Simulation	312.57	404.53	507.31	796.65628	1599.64
Prototype	406.23	515.78	636.17	941.26	1824.87

Table 5.5: Average Network Discovery Times : Lab Simulation vs. Prototype



(a) Network Discovery Times Comparison: All cases



(b) Network Discovery Times Comparison: N/8

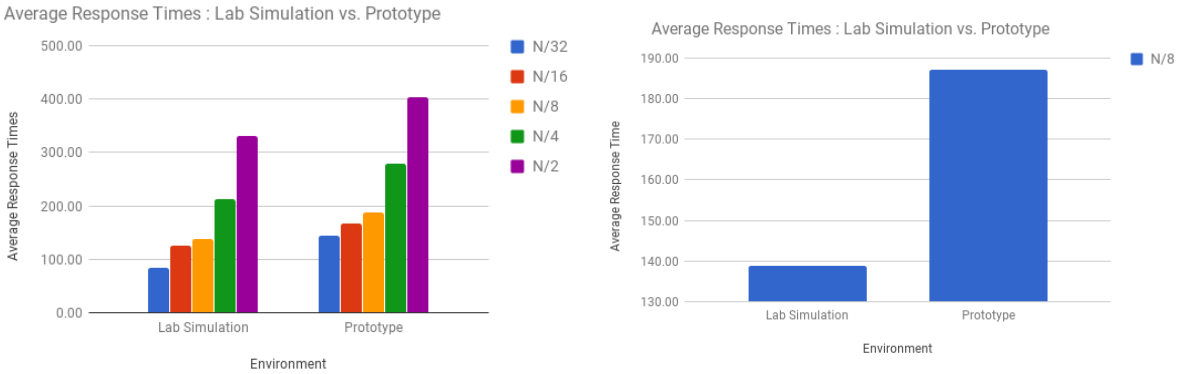
Figure 5.5: Average Network Discovery Times : Simulation Environment vs Facebook

Here we compare the average network discovery times in both the cases. We observe the following:

- Average network discovery times obtained from prototype evaluation are more than their corresponding values obtained from the lab based simulations. This observation can be linked with the overhead of network delays while using the prototype.
- Though the actual values are different, however the patterns we obtain from the two versions are consistent as can be observed from Fig 5.5.

Environment	N/32	N/16	N/8	N/4	N/2
Lab Simulation	84.71	125.57	138.794	211.97	330.53
Prototype	144.56	165.89	187.2	278.19	402.58

Table 5.6: Average Response Times : Lab Simulation vs. Prototype



(a) Average Response Times Comparison: All cases

(b) Average Response Times Comparison: N/8

Figure 5.6: Average Response Times : Lab Simulation vs. Prototype

5.4.3 Average Response Times: Lab Simulation vs Prototype

Here we compare the average response times in both the cases. We observe the following:

- Average response times obtained from prototype evaluation are more than their corresponding values obtained from the lab based simulations. This observation can be linked with the overhead of cryptography, steganography and network delays while using the prototype.
- Though the actual values are different, however the patterns we obtain from the two versions are consistent as can be observed from Fig 5.6.

We hence conclude that the results we obtain while comparing the prototype with the lab based simulation are encouraging and demonstrate the *usability* of our proposed design.

Chapter 6

Limitations and Future Work

6.1 Limitations

One common argument against our proposed system would be that the users participating in file sharing over Facebook are themselves not anonymous or pseudonymous. However, we consider this very observation as the novelty of our design. *Unlinkability of sender and receiver* and *Probable Innocence* are ensured by the underlying cryptographic framework. Further, the additional layer of *Steganography* protects the activities of our specific Facebook group from getting censored by converting the encrypted gibberish looking posts into normal looking English plain text posts. This process helps to avoid any possible suspicion or unwanted attention by making the posts look like those of any other Facebook group. Hence, despite the fact that the individual user identities are not hidden, the communications are guaranteed to be anonymous. Users can therefore still feel safe to use our proposed system while collaborating with their peers in searching and sharing of files over Facebook. We however believe that the possible limitations of our system could be due to the limitations of the components that we have chosen for our design as detailed below:

- **DropBox**: Users of our system share files with their peers by means of uploading their respective files on their personal DropBox accounts and sharing encrypted DropBox file links. However, if the DropBox links themselves reveal the identity of the creator of the document, then our entire purpose gets defeated. This issue, however can be dealt with by selecting a more secure readily available file hosting service or by developing a custom made client side file hosting service for obtaining the required secure shareable links.

- **Steganography**: Steganography is the art and science of hidden writing used to obscure the very fact that we are dealing with encrypted messages at all. In our proposed system, we are using the Steganography services provided by [3]. However, it implements a very rudimentary form of steganography by converting an encrypted message into English text consisting of words chosen from a dictionary of 65536 (2^{16}) words. This issue, however can be addressed by selecting better *Text Steganography* techniques or by exploring the domains of *Image Steganography*, *Audio Steganography* or *Video Steganography* [13].

6.2 Future Work

- In future, we propose to address the existing shortcomings of our system as highlighted in the previous sections. We would also like to investigate how AnonSocialMix defends against more kinds of attacks like the Sybil attack.
- Our current implementation makes it compulsory for the users to join a specific Facebook group in order to participate in anonymous P2P file sharing. In future releases, we would like to do away with this constraint and make the architecture more open.
- We would like to extend our proposed architecture to other Social Networks as well and evaluate their performance.
- We would also like to extend our work to media content distribution while avoiding copyright issues for both the producer and the consumer.

Chapter 7

Conclusion

Interest in the field of anonymity and anonymous peer-to-peer systems has grown rapidly due to rise in online privacy and security issues owing to constant censorship of user Internet access. In this thesis work, we highlight the importance of user privacy and anonymity in online transactions and present a brief literature review of the existing anonymous P2P file sharing protocols and systems. We find that even popular systems like Freenet and GNUnet suffer from not enough participants. We hence explore the possibility of a scheme that allows censorship-resistant P2P file sharing on top of an inherently non-anonymous non-P2P online social network platform like Facebook which readily offers us a strong user base. We present the design of such an overlay network called *AnonSocialMix* and show that users of this proposed system no longer need to hide themselves behind a mask in order to share files with their peers online. The underlying cryptographic framework ensures that the communication remains anonymous, which in our case is defined as the unlinkability of sender and receiver. We further implement a running prototype of our proposed system in the form of a Chrome Extension and compare its performance with another version of simulation code written to test the working of our design in an ideal controlled lab environment. We obtain encouraging results in the process.

Despite the risk of getting misused by people with malicious intent, anonymous networks play a critical role in supporting free speech and privacy on the Internet and must be promoted for the sake of those who might legitimately be benefited from them [17]. We hope that this work enables more and more netizens with good intent to participate in anonymous P2P file sharing over the Internet without any fear of censorship or victimization.

Bibliography

[1] *https://sites.google.com/site/cs181anonymity/definition*.

[2] *https://en.wikipedia.org/wiki/Anonymous_P2P*.

[3] *https://www.fourmilab.ch/javascrypt/stego.html*.

[4] https://prezi.com/mtrizshw2f-5/tor-presentation/.

[5] Tor network status, retrieved 2nd august 2017. *http://torstatus.blutmagie.de/*.

[6] Chothia, T., and Chatzikokolakis, K. A survey of anonymous peer-to-peer file-sharing. *Proceedings of the 2005 international conference on Embedded and Ubiquitous Computing* (December 2005), 744–755.

[7] Douceur, J. The sybil attack. *Proceedings of IPTPS* (March 2002).

[8] Festor, J. P. T. C. Improving content availability in the i2p anonymous file-sharing environment. *Cyberspace Safety and Security, Lecture Notes in Computer Science, LNCS 7672* (2012), 77–92.

[9] H. Yu, M. Kaminsky, P. B. G., and Flaxman, A. Sybilguard: defending against sybil attacks via social networks. *Proceedings of ACM SIGCOMM* (2006).

[10] Han, J., and Liu, Y. Rumor riding: Anonymizing unstructured peer-to-peer systems. *Proceedings of IEEE ICNP* (2006).

[11] Hoang, N. P., and Pishva, D. Anonymous communication and its importance in social networking. *16th International Conference on Advanced Communication Technology* (February 2014).

[12] Hong, I. C. S. W. W. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science LNCS 2009* (March 2001).

[13] Kour, J., and Verma, D. Steganography techniques âĂŞ a review paper. *International Journal of Emerging Research in Management & Technology 3* (May 2014).

[14] Michael G. Reed, P. F. S., and Goldschlag, D. M. Anonymous connections and onion routing. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS 16* (May 1998).

[15] Mittal, P., and Borisov, N. Shadowwalker: Peer-to-peer anonymous communication using redundant structured topologies. *Proceedings of ACM CCS* (2009).

[16] Reiter, M. K., and Rubin, A. D. Crowds: anonymity for web transactions. *ACM Trans. Inf. Syst. Secur. 1* (November 1998), 66–92.

[17] Rob Kling, Ya-Ching Lee, A. T., and Frankel, M. S. Assessing anonymous communication on the internet: Policy deliberations. *The Information Society 15* (April 1999), 79–90.

[18] Roger Dingledine, N. M., and Syverson., P. Tor: the second-generation onion router. *Proceeding SSYM'04 Proceedings of the 13th conference on USENIX Security Symposium 13* (August 2004), 303–320.

[19] S. Katti, J. Cohen, D. K. Information slicing: Anonymous using unreliable overlays. *Proceedings of USENIX NSDI* (2007).

[20] Stephanos Androutsellis-Theotokis, D. S. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys (CSUR) 36* (December 2004), 335–371.

[21] V. Scarlata, B. L., and Shields, C. Responder anonymity and anonymous peer-to-peer file sharing. *Proceedings Ninth International Conference on Network Protocols. ICNP* (November 2001).