Project Report

CSE 6809 - Distributed Search Techniques

Towards Sybil-resilient P2P Networks

24 October, 2010

Md. Tanvir Al Amin

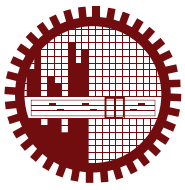
Student No. 04 09 05 2064

S M Rifat Ahsan

Student No. 10 09 05 2060

Supervised By

Dr. Reaz Ahmed



Department of Computer Science and Engineering

Bangladesh University of Engineering and Technology (BUET)

Dhaka - 1000

Abstract

Recently, Sybil attack has become one of the most visible security problems in the peer to peer networks where it is not possible to depend on trusted authorities for admission control or maintenance. In such networks or in any open distributed systems, an adversary creates many fake identities in order to increase its influence and deny service to honest participants. Defending against this attack is challenging because creating many fake identities is cheap. In this project, our contributions are two-fold. Our main contribution is a sybil-resistant Distributed Hash Table, where sybil nodes can join, but the routing algorithm for an honest node can effectively bypass the sybil identities. Our method assumes that each node have direct knowledge about their neighbors and have established social trust relationships with them. With this knowledge in hand, our methods uses gossip algorithms to assign IDs to the nodes in a distributed manner. The IDs are linear binary codes (for instance, extended Golay codes) which constitutes the XOR based routing mechanism for the distributed hash table "Plexus." Existing sybil-proof DHTs either depend on explicit ID generation, or routings take grossly suboptimal paths, or the finger table size becomes larger. However, our scheme successfully manages all of these parameters and as a added benefit of Plexus, offers disjoint diversified optimal routes and mirroring capability for fault tolerance. Based on these findings, we have also presented a layered architecture of a distributed admission control and maintenance scheme for a peer to peer network, which is our second contribution.

Contents

Chapter 1

Introduction

1.1 Sybil Attack

The name Sybil is derived from the book [1] by F. R. Schreiber, which is a tale of a patient "Sybil Dorsett," whose childhood was so agonizing that she was suffering from dissociative identity disorder and manifested sixteeen different personalities. Douceur [2] was the first to consider this multiple identity problem in the context of large scale peer to peer systems.

Large scale distributed or peer 2 peer systems, as the name implies, depend a lot on collaborative protocols and in general are vulnerable to possible malicious activity from compromised, faulty or hostile remote computing elements. However, a single faulty element may not be capable of launching detrimental attack to the distributed system. But if it is possible for it to create multiple identities, it can take control of the substantial fraction of the system and subvert it. Exploiting this security breach is commonly known as sybil attack. In his seminal paper, Douceur proved that it is not possible to resist sybils completely without any trusted central certification system.

Many forms of sybil attack had been invented thereafter. Ironically, the problem itself takes several forms depending on the context. Wherever there is a distributed system without any central authentication system, the Sybil attack is a potential candidate for the security holes. Creating sybil identities itself may not be an active attack, but can be termed as an "Attack Template." Based on the attacker model, it might even not be possible to discern real identities from sybil identities, especially when human effort is involved in Sybil's part.

Think about a user, creating multiple "free" email accounts and sending spam

to people or apply some social engineering to get some information. Such human intervened sybil behavior is less pronounced because the cost of time and energy is a bound on number of identities a human can create. In fact, this attack model does not belong to the domain of computer science, rather it is a social issue; free identities just made it easier for an attacker to hide the real identity. Lots of research has been done in this direction with some beautiful solutions like Captcha [3] or Computational puzzles [4].

An application domain where the effect of Sybil attack has been pronounced most is rating systems like PageRank [5], Content rating, Internet Polling or Reputation systems. Available software solutions like TubeAutomator [6] or FriendBomber [7] makes it very easy to greatly affect a recommendation system. [8], [9] discuss Sybil resistance in recommendation systems. [10] introduce the idea of rating history and individual trusts imitating day-to-day social behavior and provide a method for nullifying the Sybil votes in a content rating system. However, their method is based on statistical property of the votes and according to their protocol, an individual gives low emphasis or trust to users, whose historical rating on an object differs much from the individual's rating. Actually, it finds the outliers with respect to an individual user in the historical data of rating. Affect of Sybil identities has been pronounced in Online Social Networks and in Social network based applications also.

Outside the application domain, Sybil attack is highly visible in structured P2P systems or Distributed Hash Tables like Chord [11], Kademlia [12], Pastry [13] or Skipnet [14]. Malicious identities mainly take several strategies for disrupting the operation of the structured overlays. Sybil nodes can install themselves in the network, but not provide any information to the other nodes like failing to look up nodes or keeping null finger table. Another strategy can be providing another malicious node as the reply resulting in a wild goose chase. A malicious entry can provide with wrong content also, thus spoiling honest users bandwidth. On the other hand a sybil can purposefully eat up space in the DHT id space, essentially creating a Denial-of-Service attack. More importantly, it can hook itself onto "hot" regions in the ID-space to steal information. [15] et al. show results about observed Sybil attacks in Maze P2P system.

Existing DHT protocols incorporate little or no defense against Sybil attack. For example, [16] [17] and several other authors discuss how easy it is to subvert the KAD network by Sybil attack. As there is no certification system, it becomes hard to control admission of Sybil nodes into the DHT. [18] and [19] proposed

admission systems based on peer administered computational puzzles. Later, It became obvious that, effective Sybil defense caters for some external information if a trusted authority is not to be used. Protocols using social information and trust metrics were introduced. In his PhD dissertation, Levien [20] discusses trust metrics based on social networks. Marti et al. [21] made use of social network information to route messages over trusted nodes. On the other hand, Danezis et. al. [22] attempt to eliminate trust bottlenecks and provide a Sybil Resistant DHT routing model. They use the social information present in the introduction graph of the network and combine the traditional closeness parameter of Chord and a trust based route diversity parameter to diversify the routes.

Sybilguard [23] was a breakthrough paper after all these research which introduces an important observation to account for Sybil attacks in social networks. They claim that some properties of the social network is an important tool against Sybil attacks. The legitimate users are part of a social network, and direct links in the social network means that the peers have exchanged keys by some other means (like socially recognizable contacts). SybilGuard divides the network into an honest region and a sybil region. Honest region to Sybil region trusts are called attack edges and it can be easily perceived that there is a small cut along the attack edges as Sybil identities cannot create large number of trust relationships with the honest portion. Due to the fast mixing property of social graphs, a random walk in the graph follows stationary distribution, i.e. probability that the last hop resides inside the honest region is high. [23] and [24] takes into account these random walks from a suspect and a verifier, and applies some heuristics on the number of intersection nodes to take a decision about a suspect being "probably honest" accepting $O(\log n)$ Sybil nodes along an attack edge.

Some recent works have adapted the concept of random walks for purposes other than SybilLimit. Tran et al. [9] used it for Sybil-resilient content rating, Danezis and Mittal used it for SybilInfer, an inference system based on Bayesian Learning [25]. Recently, Laas and Kaashoek [26] utilized the social trust information and the fast mixing property to create random walks in the social graphs and created a one-hop DHT based on layered ID values. They trade table size to reduce hop count and make it a one-hop DHT. However the finger table size now becomes $O(\sqrt{n}\log n)$. There are some purposeful design changes they have adopted to apply the idea of random walks in a social graph to a Chord-like DHT. Like, their DHT is no longer a circular chord, based on ID values, rather ID values have an unknown ordering.

They target the sybil attack of two types, clustered attacks to purposefully get specific ID stretches into control, and random point attack.

However, Sybil attacks can be useful in some sense. Davis et. al. [27] discuss one use of Sybil attack in fighting Botnets. Botnets provide attackers with the large scale low cost computing infrastructure required to engage in major spam campaigns, larger-scale phishing attacks, etc. Over time, botnets have evolved toward using decentralized peer-to-peer systems. While subsequent amount of P2P research is based on making the DHT sybil resist, the study in [27] proves to be useful against botnets, which also proves how insecure traditional DHT systems are.

1.2 Our Proposal

Main outcome of our research is a sybil resistant routing scheme, which will work as the basis DHT for the peer 2 peer application. We use novel linear binary code based ID and some properties of social networks to create a distributed hash table. While other solutions either increase the finger table size significantly or increases number of routing hops to provide sybil resilience, our proposed scheme successfully keeps both of these crucial performance parameters in control. It treats each community of a social network independently. While other sybil-resistant routing schemes just work between one honest region and one sybil region, our protocol is capable to work in the more practical case where there are multiple communities. Moreover, this method neither assumes any unrealistically large table size, nor it makes the routes suboptimal.

We have also proposed a sybil resistant scheme for a peer 2 peer application like file sharing. Our scheme adopts a layered architecture for such an application. It divides the job of admission control and maintenance in these layer and uses some cross layer information also.

Chapter 2

Social Graphs

It is already evident that information hidden in a social graph is important for sybil resilience. SybilGuard and other algorithms in face exploit some of those properties. Here we discuss some properties of social graphs. An example social network is shown in Figure 2.1

2.1 Scale Free Networks

Social Networks are scale free. A scale-free network is a network whose degree distribution follows a power law, at least asymptotically. That is, the fraction $P(k)$ of nodes in the network having k connections to other nodes goes for large values of k as $P(k) \sim k^{-\gamma}$, where γ is a constant whose value is typically in the range $2 < \gamma < 3$, although occasionally it may lie outside these bounds.

Scale-free networks are noteworthy because many empirically observed networks appear to be scale-free, including the world wide web, the Internet, citation networks, and some social networks. [28]

As with all systems characterized by a power law distribution, the most notable characteristic in a scale-free network is the relative commonness of vertices with a degree that greatly exceeds the average. The highest-degree nodes are often called "hubs," and are thought to serve specific purposes in their networks, although this depends greatly on the domain.

The power law distribution highly influences the network topology. It turns out that the major hubs are closely followed by smaller ones. These ones, in turn, are followed by other nodes with an even smaller degree and so on. This hierarchy allows for a fault tolerant behavior. Since failures occur at random and the vast

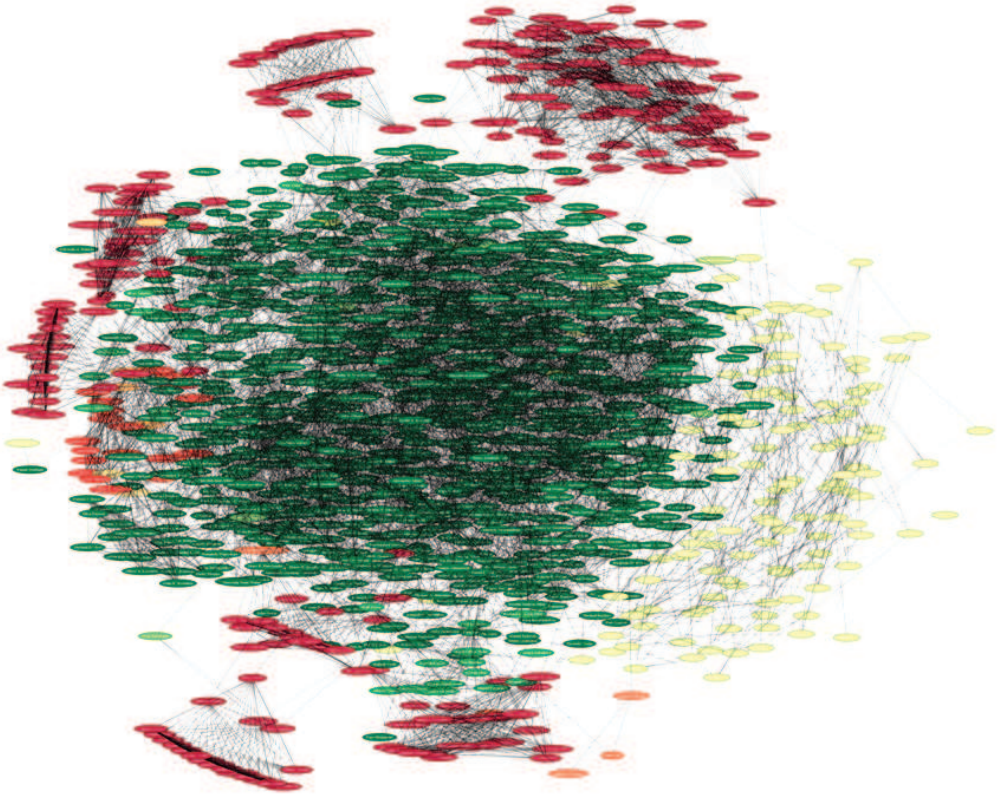


Figure 2.1: Example of DBLP social graph

majority of nodes are those with small degree, the likelihood that a hub would be affected is almost negligible. Even if such event occurs, the network will not lose its connectedness, which is guaranteed by the remaining hubs. On the other hand, if we choose a few major hubs and take them out of the network, it simply falls apart and is turned into a set of rather isolated graphs. Thus hubs are both the strength of scale-free networks and their Achilles' heel.

Another important characteristic of scale-free networks is the clustering coefficient distribution, which decreases as the node degree increases. This distribution also follows a power law. That means that the low-degree nodes belong to very dense sub-graphs and those sub-graphs are connected to each other through hubs. Consider a social network in which nodes are people and links are acquaintance relationships between people. It is easy to see that people tend to form communities, i.e., small groups in which everyone knows everyone (one can think of such community as a complete graph). In addition, the members of a community also have a few acquaintance relationships to people outside that community. Some people, however, are so related to other people (e.g., celebrities, politicians) that they are connected to a large number of communities. Those people may be considered the hubs responsible for the small world phenomenon.

At present, the more specific characteristics of scale-free networks can only be discussed in either the context of the generative mechanism used to create them, or the context of a particular real-world network thought to be scale-free. For instance, networks generated by preferential attachment typically place the high-degree vertices in the middle of the network, connecting them together to form a core, with progressively lower-degree nodes making up the regions between the core and the periphery. Many interesting results are known for this subclass of scale-free networks. For instance, the random removal of even a large fraction of vertices impacts the overall connectedness of the network very little, suggesting that such topologies could be useful for security, while targeted attacks destroys the connectedness very quickly. Other scale-free networks, which place the high-degree vertices at the periphery, do not exhibit these properties; notably, the structure of the Internet is more like this latter kind of network than the kind built by preferential attachment. Indeed, many of the results about scale-free networks have been claimed to apply to the Internet, but are disputed by Internet researchers and engineers.

As with most disordered networks, such as the small world network model, the average distance between two vertices in the network is very small relative to a

highly ordered network such as a lattice. The clustering coefficient of scale-free networks can vary significantly depending on other topological details, and there are now generative mechanisms that allow one to create such networks that have a high density of triangles.

2.2 Metrics in Social Network Analysis

Here we discuss important metrics pertinent to computational sociology or social network analysis, which we will eventually use for sybil-proof DHT routing.

Betweenness The extent to which a node lies between other nodes in the network. This measure takes into account the connectivity of the node's neighbors, giving a higher value for nodes which bridge clusters. The measure reflects the number of people who a person is connecting indirectly through their direct links.

Bridge An edge is said to be a bridge if deleting it would cause its endpoints to lie in different components of a graph.

Centrality This measure gives a rough indication of the social power of a node based on how well they "connect" the network. "Betweenness", "Closeness", and "Degree" are all measures of centrality.

Closeness The degree an individual is near all other individuals in a network (directly or indirectly). It reflects the ability to access information through the "grapevine" of network members. Thus, closeness is the inverse of the sum of the shortest distances between each individual and every other person in the network. The shortest path may also be known as the "geodesic distance".

Cohesion The degree to which actors are connected directly to each other by cohesive bonds. Groups are identified as cliques if every individual is directly tied to every other individual, social circles if there is less stringency of direct contact, which is imprecise, or as structurally cohesive blocks if precision is wanted.

Degree The count of the number of ties to other actors in the network.

Eigenvector Centrality A measure of the importance of a node in a network. It assigns relative scores to all nodes in the network based on the principle that

connections to nodes having a high score contribute more to the score of the node in question.

2.3 Distinctive properties of social networks

Social networks have distinctive empirical properties.

- Dunbar's number limits the number of stable social relationships a user can have to less than a couple of hundred. It is linked to size of neo-cortex region of the brain. It is observed throughout history since hunter-gatherer societies and is roughly reported to be 150.
- Major social networks all have short path length from 4.25 5.88
- the graphs have a densely connected core comprising of between 1% and 10% of the highest degree nodes such that removing this core completely disconnects the graph. This core links small groups of strongly clustered, low-degree nodes at the fringes of the network.
- Social networks are fast mixing. This property is used by SybilGuard and later algorithms.They exploit this property by a random walk. The mixing time of the network is the number of hops k required so that probability of staying at any node becomes uniform. In a fast mixing graph, this value is $O(\log n)$ meaning, only $O(\log n)$ hops are required so that getting to any node becomes equiprobable.
- Creating social links or trust relationships involve significant human effort.

Chapter 3

Sybil-proof DHT

At the heart of our proposed scheme lies the sybil-resistant DHT, based on Plexus [29]. Plexus has a partially decentralized architecture involving superpeers. It adopts a novel structured routing mechanism derived from the theory of Error Correcting Codes (ECC). Plexus achieves better resilience to peer failure by utilizing replication and redundant routing paths. Routing efficiency in Plexus scales logarithmically with the number of superpeers. The concept presented in this paper is supported with theoretical analysis, and simulation results obtained from the application of Plexus to partial keyword search utilizing the extended Golay code.

3.1 Social networks vs Sybil networks

The social network is an undirected graph whose nodes know their immediate neighbors.

Figure 3.1 conceptually divides the social network into two parts, an honest region containing all honest nodes and a Sybil region containing all Sybil identities. An attack edge is a connection between a Sybil node and an honest node. An honest edge is a connection between two honest nodes. An "honest" node whose softwares integrity has been compromised by the adversary is considered a Sybil node. The key assumption is that the number of attack edges, g, is small relative to the number of honest nodes, n. As pointed out by earlier work, one can justify this sparse cut assumption by observing that, unlike creating a Sybil identity, creating an attack edge requires the adversary to expend social-engineering effort: the adversary must convince an honest person to create a social link to one of its Sybil identities.

The correctness of our algorithm will depend on the sparse cut assumption,

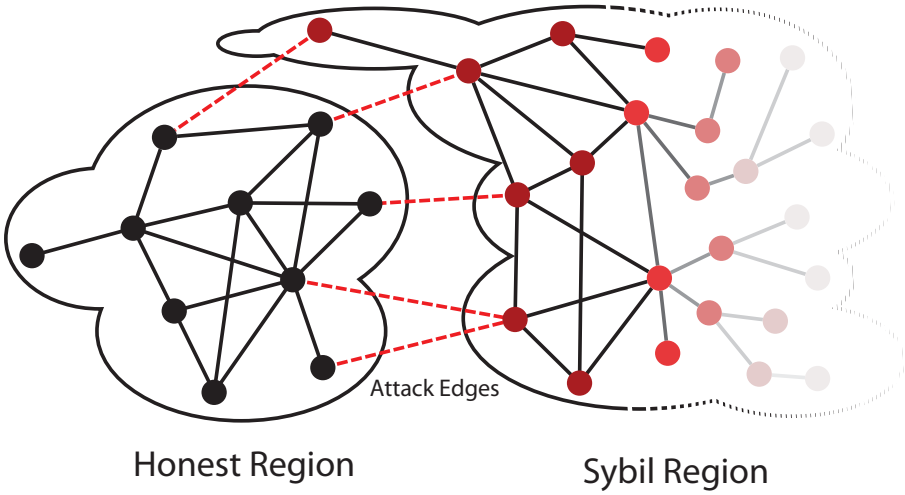


Figure 3.1: The social network. A sparse cut (the dashed attack edges) separates the honest nodes from the Sybil nodes. The Sybil regions size is not well-defined, since the adversary can create new pseudonyms at will.

but its performance will not depend at all on the number of Sybils. In fact, the protocol is totally oblivious to the structure of the Sybil region. Therefore, the classic Sybil attack, of creating many fake identities to swamp the honest identities, is ineffective. Since we rely on the existence of a sparse cut to distinguish the honest region from the Sybil region, we also assume that there is no sparse cut dividing the honest region in two. Given this assumption, the honest region forms an expander graph. Expander graphs are fast mixing, which means that a short random walk starting from any node will quickly approach the stationary distribution. Roughly speaking, the ending node of a random walk is a random node in the network, with a probability distribution proportional to the nodes degree. The mixing time, w, is the number of steps a random walk must take to reach this smooth distribution. For a fast mixing network, $w = O(\log n)$.

3.2 Design Goals

DHT like Whanau has kept no specific ordering of node IDs. Each of the nodes keep a finger table of size $O(\sqrt{n}\log n)$. The IDs are no longer along any specific ordering and the DHT has become a one-hop DHT. On the other hand, there are ways like diversifying the routes along with social information, which are based on Chord, but the sybil-resistant routes are far from optimal.

As social networks are groups of closely knit communities and trust relationships, it is hard for a sybil to create too many attack edges. Hence there is a low cut between the honest community and sybil zone. However, we prefer to create the DHT, so that the routing between two nodes resembles the social path between them. Hence we want to build a DHT Where distance between two nodes in the DHT-Space is related to their social-distance i.e, two friends in the social graph are expected to be one-hop distant in the DHT-Space Most of the queries will be through friends Hence, the probability of reaching a Sybil node is less.

3.3 Plexus

Plexus uses hamming distance based linear binary codes as IDs. These ID values are based on golay codes.

3.3.1 Plexus Routing

Consider a (n, k, d) linear code (C) with generator matrix $G_C = [g_1, g_2, ..., g_k]^T$. To route using this code, plexus peer X has to maintain links to $(k+1)$ superpeers with IDs X_1, X_2, ... X_{k+1} as follows:

$$X_i = \begin{cases} X \oplus g_i & 1 \leq i \leq k \\ X \oplus g_1 \oplus g_2 \oplus \ldots \oplus g_k & i = k+1 \end{cases}$$

In such an overlay, it is possible to route a query from any source to any destination codeword in less than or equal to $frack2$ routing hops [29]. Thus, if extended Golay code, (24, 12, 8) self dual linear binary code is used, then it is always possible to route to any node in 6 hops. As $2^{12} = 4096$ codewords exist, we can suffice 4096 IDs.

3.4 Observations

- In general, social networks contain many loosely interconnected cores.The cores are densely connected. For the sake of present work, we suppose that there is only one such real community in the network, and the other community is the sybil zone or phony identities.

- Each community in the social network is fast mixing.

3.5 Extending Plexus for Sybil-proof DHT

Here we describe our protocol for creating a Plexus DHT out of a social network.

- Suppose there are 2^k communities in the social network, each having 2^k nodes. If extended Golay codes are used then the code is (24, 12, 8) and $k = 12$. However, other codes with linear binary properties and XOR based routing metric can be used as well.

- Our protocol essentially makes IDs two level. For each community, there is one ID. Nodes of a community will knows that ID. However each community has a separate codespace, for each node inside that community, there is another ID from that codespace.

- First step is to identify the communities in a distributed manner. Figure 3.2 shows a social network where more that one honest communities exist.

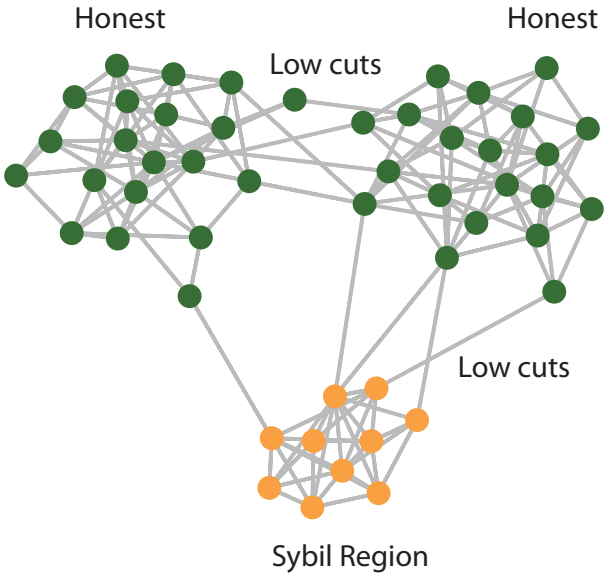


Figure 3.2: A social network with two honest regions and one sybil region

- All nodes u know their friends $F_1(u)$. All nodes u send $F_1(u)$ to all of their friends. bandwidth cost = $O(deg(u))$ for each node u.

- At this point, every node u, in addition to $F1(u)$, can calculate its "mutual friend list" for each of its friends. For any two friends u, v : $M_1(u, v) = F_1(u) \cap F_1(v)$ is their mutual friend set.

- Every node u, can also calculate $F_2(u)$, its exact two-hop distant friend list. $F_2(u) = F_1(F_1(u)) - F_1(u) - u$

- Thus, every node exactly knows the following information about the topology around it

  - Its own friends
  - Two hop friends
  - Connections between one-hop friends.
  - Connections between one-hop and two-hop friends.
  - However, it doesn't know about any connection between two-hop friends. This is, realistic in a society.

- Each node u calculates a metric to have a trust value on each of its friends v. The metric can be based on $M_1(u, v)/F_1(v)$ i.e. what fraction of v's friends are u and v's mutual friends. This metric can also be based on other formula also, the exact form of which needs experimentation. This way, attack edges are prevented.

- After this step, we use distributed leader selection algorithms to elect a temporary node to generate the first ID in that community. Instead of distributed leader selection algorithms, other schemes can be employed also.

- After the first ID L is generated, the leader node sends this ID to each of its friends. As each node maintains a finger table and the generator matrix, each can now generate the IDs for the immediate friends of the leader L via $L \oplus g_1$ or $L \oplus g_2$ and so on.

- Each neighbor selects an ID using the XOR of L and some g_s. After selected and ID, it sends thsi message to its trusted friends. There can be collision also. But after some levels of iteration, these collisions can be settled down.

- After all the nodes in a community has picked up an ID, that community is over. Provided the trusted edges are used, results in possible filtering out of

the sybil nodes. As it is easy to create sybil nodes but hard to create social links, most of the sybil (or community non-member) are prevented this way.

- Each of the temporary leaders communicate via gossip algorithms [30] [31], and learns about the IDs of their community. These supernodes form the inter-community routing. On the other hand, intra-community routing is efficiently performed by Plexus as we have already assigned linear binary code for community members.

Chapter 4

Sybil Resistant P2P Application

In this chapter we propose our Architecture of a Sybil Resistant P2P Application.Primarily, we don't want to change the basic structure of a DHT to apply security patches in it.

4.1 Division into Layers

For a P2P application, our idea is to divide the responsibility in three layers.

1. Application Layer
2. DHT Layer
3. Network-Access Layer

The bottom-most layer is an "Network-Access Layer," which is basically a small application on some transport mechanism, responsible for actual packet transmission using an internet connection. The transport mechanism can be actual TCP/IP Transport Layer, or can be anything else which establishes packet sending and receiving capability over a cloud / web / messenger services.

The layer on top of it is a "DHT Layer," which is responsible for maintaining the overlay connectivity and maintenance, distributed database, setup and lookup procedures.

The topmost layer is the "Application Layer," which is the main application, dealing with the application logic and policy. There are provisions for cross layer feedback between DHT Layer and Application layers for security purposes. It is the

Application layer which controls the "social-trust" graph. Friendship also depends on it.

Security is imposed via following mechanism

1. Admission control at Application layer
2. Social trust, Friendship, controlled at Application layer
3. Application Object (Files or other shared items) rating and Rating behavior recording (determined at application layer)
4. Routing behavior rating, Query reply rating and rating behavior recording (determined at DHT Layer)

Our application level idea is similar to evolution in human society. A new participant neither become part of the DHT, nor it gets any identifier for the DHT. Rather it is connected at the social graph, to some other participant. For the time being, the introducer participant will be known as its "parent." The new participant should have social trust / friend relationship with that introducer participant. It can send friend requests to other nodes as well, and that nodes can decide to accept or reject the request using their own discretion. They should not accept an unknown node as a friend, rather they should accept according to their offline social relationships, i.e. they should accept only if there is a social trust.

An new participant have certain limited capabilities. It will have full capability, when it becomes a "First class citizen" and inserted into the DHT. To do that, it has to gain enough trust / friends from the rest of the society. The process is just like that : a new node is introduced with the society, and it tries to mingle more and more with social relationships. (May be, it can be introduced into the DHT at that time, when it is no longer detected it as a sybil. Suspect-Verifier test, but how to verifier node will be selected ? And the test itself has some problems actually)

A new node

1. Can not perform any DHT lookup, or can not perform in any DHT level ratings.
2. May or may not perform Application object ratings, depending on application policy.
3. May not have full permissions capabilities at application (depending on the application policy).

4. Can not make social relationships with another "new" or "immature" node

However, merely keeping a new node out of application layer service is not a good idea, because it will complicate the growth of the "society." To enable the new or "child" nodes use the P2P service, the introducer/parent node have to work as a proxy for it until it becomes a DHT member.

Explanation of the lookup process for immature nodes: Any friend of an "immature" node should work as the proxy for it (only for DHT lookup process). Content object / Shared files will be exchanged directly, but as the immatured nodes can not have acess to DHT. All lookup for them will be done by its matured friends. It may want to load balance the queries among the friends. And the content / files shared by the child, will also be represented by their parents / friends. Any lookup of that content should return the IP address of a friend / parent. However, the friend / parent will then provide the IP address of the child. Then the file transfer can work directly. However, based on the behavior of its provided content, first class citizens / DHT members will rate it. If the new node gets bad rating, its probability of becoming a DHT member will be less. Process of becoming a first class citizen is to be determined by the application.

Inside the DHT a database will be maintained. Each record of the database can contain fields like ID, IP Address, Routing rating, Content rating, Behavior vector etc.

4.2 Sybil attack based on attacker type

For our sybil resistant peer 2 peer architecture, we discuss the cases based on the attacker type.

- A new node wants to launch a sybil attack. This is the trivial case. As a new node is not inserted into DHT, and is functioning via a proxy, it is not possible for a new node to launch sybil attack.

- A matured node (member of DHT, first class citizen) wants to launch a sybil attack.

- A group *RG* of socially connected matured nodes become compromised and they want to subvert the system by helping each others in group to create a "sybil society" against the original society. Hence they create sybil identities

and each of them accept other's childs as friends creating a pseudo society of mixed sybils and reals. (possible solution : A civil war, and a break up of the underlying DHT in two parts. No other protocol has handled this issue before, however, it should be possible to extend SybilGuard to divide the social graph in two parts using low cuts, but with decreased sybils per attack edge, like $O(\log n/|RG|)$

- Somehow, a dishonest node has entered the DHT.
- What will happen when a node has acquired a place in the DHT, and have captured a lot of data (id values upto next node) and now it has become compromised and refuse to give back the data it was saving. In this way, lots of sybil either clustered or spread throughout the DHT ring and is now refusing to give data they are saving.

Chapter 5

Conclusions

Main success of our Sybil-resilient DHT is, it treats each community independently. While other sybil-resistant routing schemes just work between one honest region and one sybil region, our protocol is capable to work in the more practical case where there are multiple communities. Moreover, this method neither assumes any unrealistically large table size, nor it makes the routes suboptimal.

As an added benefit from Plexus, our method can utilize disjoint optimal routes. Hence, sybil resilience can be further extended by using diversified routing, without increasing the path length. Moreover, as Plexus accounts for mirrored nodes, our method provides a mirroring scheme also, further decreasing the damage caused by sybil nodes, about which the previous protocols remained silent.

We want to examine some theoretical properties of our protocol about how robust it is or how effectively it can work against sybil identities. Providing rigorous mathematical proofs and theoretical bounds are one of our next goals. Along with that, we are also planning for experimenting with our protocol thoroughly and fine tuning the parameters. Several issues are still unresolved which needs to be solved. Those issues include the selection of leader nodes, proper gossip algorithms, and selecting the trust metric.

Bibliography

[1] F. R. Schreiber, *Sybil*, 1973.

[2] J. Douceur and J. S. Donath, "The sybil attack," in *IPTPS*, 2002, pp. 251–260.

[3] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford, "Captcha: Using hard ai problems for security," in *EUROCRYPT*. Springer-Verlag, 2003, pp. 294–311.

[4] A. Juels and J. Brainard, "Client puzzles: A cryptographic countermeasure against connection depletion attacks," in *Proceedings of NDSS '99 (Networks and Distributed Security Systems)*, S. Kent, Ed.

[5] Wikipedia Contributors, Wikipedia– The Free Encyclopedia, "Pagerank." [Online]. Available: http://en.wikipedia.org/wiki/PageRank

[6] "Youtube automator." [Online]. Available: http://www.tubeautomator.com/

[7] "Friend bomber." [Online]. Available: http://www.stealthfriendbomber.com/

[8] K. Walsh, "Experience with an object reputation system for peer-to-peer filesharing," in *In USENIX NSDI*, 2006, pp. 1–14.

[9] N. Tran, B. Min, J. Li, and L. Subramanian, "Sybil-resilient online content voting," in *In Proceedings of the 6th Symposium on Networked System Design and Implementation (NSDI)*, 2009.

[10] H. Yu, C. Shi, M. Kaminsky, P. B. Gibbons, and F. Xiao, "Dsybil: Optimal sybil-resistance for recommendation systems," Tech. Rep., 2009.

[11] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for internet applications," in *ACM SIGCOMM*, 2001, pp. 149–160.

[12] P. Maymounkov and D. Mazires, "Kademlia: A peer-to-peer information system based on the xor metric," in *Proc. IPTPS*, 2002.

[13] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," *IN: MIDDLEWARE*, pp. 329–350, 2001.

[14] N. J. A. Harvey, M. B. Jones, S. Saroiu, M. Theimer, and A. Wolman, "Skipnet: A scalable overlay network with practical locality properties," 2003.

[15] "An empirical study of collusion behavior in the maze p2p file-sharing system," 2007.

[16] P. Wang, J. Tyra, E. Chan-tin, T. Malchow, D. F. Kune, and Y. Kim, "Attacking the kad network."

[17] M. Steiner, T. En-najjary, and E. W. Biersack, "Exploiting kad: Possible uses and misuses," *ACM SIGCOMM CCR*, vol. 37, p. 2007.

[18] H. Rowaihy, W. Enck, P. Mcdaniel, and T. L. Porta, "Limiting sybil attacks in structured peer-to-peer networks," Tech. Rep., 2005.

[19] N. Borisov, "Computational puzzles as sybil defenses," 2006.

[20] R. Levien, "Attack resistant trust metrics," Ph.D. dissertation, U.C. Berkeley.

[21] P. Ganesan and H. G. Molina, "Sprout: P2p routing with social networks," in *In First International Workshop on Peer-to-Peer and Databases (P2P&DB 2004)*, March 2004.

[22] G. Danezis, C. Lesniewski-laas, M. F. Kaashoek, and R. Anderson, "Sybil-resistant dht routing," in *In ESORICS*. Springer, 2005, pp. 305–318.

[23] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "Sybilguard: Defending against sybil attacks via social networks," in *In ACM SIGCOMM 06*. ACM Press, 2006, pp. 267–278.

[24] H. Yu and M. Kaminsky, "Sybillimit: A near-optimal social network defense against sybil attacks."

[25] G. Danezis and P. Mittal, "Sybilinfer: Detecting sybil nodes using social networks."

[26] C. Lesniewski-laas and M. F. Kaashoek, "Whanau: A sybil-proof distributed hash table," in *In NSDI*, 2010.

[27] C. Davis, J. M. Fernandez, and S. W. Neville, "Optimizing sybil attacks against p2p-based botnets," 2009.

[28] Wikipedia Contributors, Wikipedia– The Free Encyclopedia, "Scale free networks." [Online]. Available: http://en.wikipedia.org/wiki/Scale-free_network

[29] R. Ahmed and R. Boutaba, "Plexus: A scalable peer-to-peer protocol enabling efficient subset search," *IEEE/ACM Transactions on Networking*, vol. 17, no. 1, February 2009.

[30] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information." IEEE Computer Society, 2003, pp. 482–491.

[31] M. Jelasity and A. Montresor, "Epidemic-style proactive aggregation in large overlay networks," in *In Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS04*. IEEE Computer Society, 2004, pp. 102–109.