Tolerating Denial-of-Service Attacks Using Overlay Networks - Impact of Topology

Ju Wang
Department of Computer Science and Engineering
University of California, San Diego
jwang@cs.ucsd.edu

Linyuan Lu
Department of Mathematics
University of California, San Diego
llu@math.ucsd.edu

Andrew A. Chien
Department of Computer Science and Engineering
University of California, San Diego
achien@ucsd.edu

ABSTRACT

Proxy-network based overlays have been proposed to protect Internet Applications against Denial-of-Service attacks by hiding an application's location. We study how a proxy network's topology influences the effectiveness of location-hiding. We present two theorems which quantitatively characterize when proxy networks are robust against attacks (attackers' impact can be quickly and completely removed), and when they are vulnerable to attacks (attackers' impact cannot be completely removed). Using these theorems, we study a range of proxy network topologies, and identify those topologies favorable for location-hiding and resisting Denial-of-Service attacks. We have found that popular overlay network topologies such as Chord [25], which has been suggested for location-hiding, is in fact not a favorable topology for such purposes; we have also shown that CAN [21], a less popular overlay network, can be a good topology for location-hiding. Our theoretical results provide a set of sound design principles on proxy networks used for location-hiding.

1. INTRODUCTION

Denial-of-Service (DoS) attacks are a major security threat to Internet applications. Since 1998, there have been a series of large-scale distributed DoS attacks which effectively shut down popular sites such as Yahoo! and Amazon and the White House website was forced to move to a different location [28, 12, 4, 3, 5]. These attacks have serious economic impact and political repercussions, and may even threaten critical infrastructures and national security [14, 20, 23]. How to effectively defend against these attacks is an open and important research problem.

In a DoS attack, attackers can make the victim application unavailable to legitimate users by overloading the application with a flood of network traffic or large amount of work. DoS attacks can be categorized as *infrastructure-level* or *application-level attacks*. Figure 1 shows a typical Internet application deployment. The application service runs on a set of interconnected hosts, which is the service infrastructure; users access it via the Internet. Infrastructure-level attacks attack the service infrastructure directly, for example, by sending packet floods to saturate the victim network. Application-level attacks cause denial-of-service by requesting large amounts of work at the application level or by exploiting weaknesses in the application.

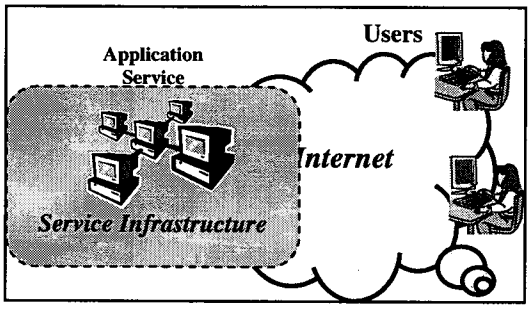

Figure 1: Internet Application Deployment.

Because many Internet applications are publicly accessible, they are easy targets for infrastructure-level DoS attacks. Many researchers are exploring the use of overlay networks to tolerate infrastructure-level attacks [26, 17, 24]. The key idea is to hide applications behind a proxy overlay network (Figure 2). Users can access applications via the proxy network without knowing their IP addresses, but attackers cannot easily locate the applications to launch attacks.

Location-hiding is an important component of a complete solution to DoS attacks. It gives applications the capability to hide their IP addresses, thereby preventing infrastructure-level DoS attacks, which depend on the knowledge of their victims' IP addresses. In general, location-hiding schemes provide a "safety period" for applications, during which applications' location is securely hidden and infrastructure-level DoS attacks are prevented. When combined with other mechanisms, such as application reconfiguration, redeployment, or even mobility, they can effectively protect applications against infrastructure-level DoS attacks. In particu-

SSRS '03, October 31, 2003, Fairfax, Virginia, USA

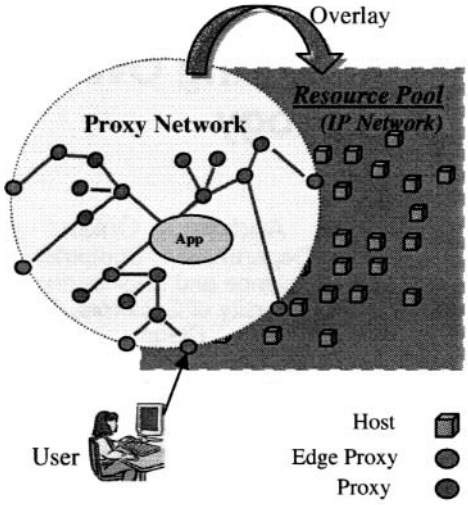

Figure 2: Proxy Network Scheme.

lar, if applications can change their location within a safety period, they can avoid DoS attacks indefinitely. Effective location-hiding schemes provide long safety periods, reducing the frequency of application reconfigurations, therefore provide strong and efficient defense against DoS attacks. It motivates our research looking for effective location-hiding schemes.

We study the proxy network described above, which is a general overlay network approach for location-hiding. Our analysis of proxy network capabilities can provide better understanding and design guidelines for the whole class of location-hiding approaches.

Proxy networks defend against attacks by means of dynamic system reconfigurations. However, the effectiveness of such schemes heavily relies on proxy network topologies. But there has been no good understanding of how to design appropriate topologies, and no way to measure the goodness of a topology. Therefore, in this paper, we extensively study the impact of proxy network topologies on location-hiding; we provide some methods to evaluate proxy network topologies, and provide some insights on how they should be designed. In particular, we provide two theorems to evaluate whether a topology is favorable for proxy networks. One theorem describes a sufficient condition for proxy networks to be robust, in which case attackers' impact can be quickly removed; the other theorem describes a sufficient condition for proxy networks to be vulnerable, in which case attackers' impact will quickly become non-trivial and cannot be easily removed.

Our results influence proxy network design in two ways: First, they serve as a screening tool to evaluate proxy network topologies, identify the favorable and filter out the undesirable. Second, they provide a set of principles one should observe during the design of proxy networks: proxy networks with high average vertex degrees are in general unfavorable, because it is hard to make them robust; furthermore, proxy networks should avoid large clusters of tightly connected nodes, because such clusters are vulnerable to attacks. On the other hand, graphs with low average vertex degrees and balanced distribution of connectivity are in general good candidates for proxy networks; because they are reasonably easy to become robust and they do not have vulnerable regions to harbor attackers' impact.

The remainder of the paper is structured as follows. Section 2 formulates the DoS problem and introduces our analytical model. Section 3 presents our analytical results, which characterize the robustness and vulnerability properties of the proxy network topology; we also present our case study on a set of network topologies, and draw some design principles for proxy networks. Section 4 presents mathematical proofs of these results. Section 5 discusses related work, and finally, we summarize in Section 6 and describe directions for future work.

2. SYSTEM MODEL

In this section, we give an overview of the proxy network scheme, describe the key components of the system, including the resource pool, the proxy network, the attacks and the related defensive mechanisms. Then we introduce an analytical model to characterize these components. Finally, we define the problem under study.

2.1 Proxy Network Scheme

Infrastructure-level DoS attacks target at the IP addresses of the victim applications. Today's Internet applications publish their IP addresses (for example via DNS), so that users can easily access the applications via the Internet. At the same time, however, their published IP addresses become obvious targets in DoS attacks. To address this problem, we use a proxy network to prevent such infrastructure-level DoS attacks by hiding the IP addresses of the applications; all accesses to the applications are mediated by the proxy network.

In our proxy network scheme (Figure 2), applications do not publish their IP addresses. Instead, they hide behind a proxy network, an overlay network that runs on a resource pool of Internet hosts. A proxy network hide the IP addresses of all the nodes inside (including internal proxies and applications); only the proxies at the edge (edge proxies) publish their IP addresses (See Figure 2). Users can only access the applications by contacting these edge proxies. No one can easily discover the IP addresses of the applications, thereby preventing infrastructure-level DoS attacks.

There are two key challenges in the proxy network scheme. First, the proxy network should hide applications' IP addresses securely. Second, the proxy network itself should be resilient to DoS attacks, so it can shield the applications. Here we focus on the first problem. It has been proved that proxy network topologies may qualitatively change the effectiveness of location-hiding [26]. However, there has been no good understanding of how to design appropriate topologies, and no way to measure the goodness of a topology. Therefore, in this paper, we extensively study the impact of proxy network topologies on location-hiding; we provide some tools to evaluate proxy network topologies, and provide some insights on how they should be designed.

2.2 Resource Pool and Proxy Network

Before discussing the attacks and the defensive mechanisms, we formally describe the resource pool and the proxy network, and introduce a rigorous terminology.

The resource pool consists of hosts in the Internet. We assume that the hosts can communicate directly if they have each other's IP addresses, and each host is identified by a unique IP address. A *node* in the overlay network is either a proxy or an application. When a node runs on a host, that host (or its IP address) is called the *location* of the node. We assume each node has a unique location at any moment (an injective mapping from nodes to hosts). Two nodes are *adjacent* if and only if they know each other's location. Obviously, adjacent nodes can communicate directly through the underlying hosts at the IP level. We use a graph (*topology graph*) to represent the overlay network. Vertices in the graph correspond to nodes in the overlay; edges correspond to the adjacency relationship. A conceptual view of a proxy network is shown in Figure 2. The topology graph describes the connectivity of the overlay network; two nodes can communicate at the overlay level if there is a path between them in the topology graph. More importantly, the topology graph also describes how the location information is shared among the overlay nodes, a critical aspect of how securely the proxy network can hide applications' location, because when attackers compromise a proxy node, they can locate all the adjacent nodes.

2.3 Attacks

We consider *host compromise* attacks, a key threat to location-hiding schemes. Host compromise attacks can penetrate proxy networks and reveal the location of the overlay nodes, including the applications. Other attacks are discussed in [26]. In a successful host compromise attack, attackers can temporarily control the victim host and steal information from it. A host under such impact is considered *compromised*, otherwise it is *intact*.

An overlay node is in one of the three states: *intact*, *exposed* and *compromised* [1]. A node is *exposed* if its location is known to attackers; therefore it is subject to future host compromise attacks. A node is *compromised* if it runs on a compromised host. Since attackers can steal information from compromised node, including the location information of all its adjacent nodes (neighbors), all the nodes adjacent to a compromised node are exposed. A node is *intact* if it is neither exposed nor compromised.

Attackers can penetrate the proxy network by repeatedly compromising exposed nodes, grow the population of compromised and exposed overlay nodes, and may eventually expose the application, thereby defeating the proxy network scheme.

2.4 Defensive Mechanisms

We have two defensive mechanisms, resource recovery and proxy network reconfiguration. Resource recovery mechanisms can recover compromised hosts into the intact state, removing attackers' impact. Examples of resource recovery include recovery or removal of infected software components on a compromised host, clean reload of system images, revocation of suspected user accounts, and so on. Proxy network reconfiguration mechanisms can dynamically change the location of proxies or the structure of the proxy network, thereby removing attackers' impact and invalidating the location information exposed to attackers. The reconfiguration mechanisms can convert exposed and compromised overlay nodes into the intact state. An example of proxy network reconfiguration is proxy migration: proxies change locations so that exposed and compromised proxies can move to intact hosts whose location is unknown to attackers [2].

Resource recovery mechanisms help to maintain sufficient intact hosts in the resource pool for proxy networks to operate, while proxy network reconfiguration mechanisms convert compromised and exposed proxies back to intact, thereby removing attackers' impact in the proxy network. These mechanisms together can shrink the size of the compromised and the exposed node population, therefore defend against host compromise attacks. We know that when there are sufficient intact hosts in the resource pool, the effectiveness of location-hiding qualitatively depends on proxy network reconfigurations [26] and resource recovery does not have significant impact in that scenario. In this paper, we assume sufficient intact hosts in the resource pool and focus on proxy network reconfiguration mechanisms. Analysis of resource recovery mechanisms can be found in [26].

2.5 Analytical Model

We model the impact of attackers and defensive mechanisms. As stated in the previous section, we focus on proxy network reconfiguration as the main defensive mechanism, and do not explicitly model resource recovery mechanisms. We assume that there exist appropriate resource recovery mechanisms to ensure sufficient resource for proxy networks.

Model $M(G, \alpha, \beta, \gamma)$: Let G be the topology graph of the proxy network defined in Section 2.2. G is a simple undirected graph. At any time t, every vertex in G is in one of the three states — intact, exposed, and compromised. We study a discrete time stochastic process, in which vertices change their state simultaneously at the end of each time step according to the following rules (Figure 3):

1. With probability α, an exposed vertex can be changed into the compromised state at the next step.
2. With probability β, a compromised vertex can be changed into the intact state at the next step (or exposed according to the last rule).
3. With probability γ, an exposed vertex can be changed into the intact state at the next step (or stay exposed according to the last rule).
4. u and v are vertices of G. If uv is an edge in G and u is compromised and v is intact, then v is instantaneously

[1] Terms "intact" and "compromised" are overloaded for overlay nodes and hosts.

[2] We assume that the proxies do not have any exploitable bugs, so that host compromise attacks can only compromise the hosts, but not the proxies. Therefore, when a proxy moves from a compromised host to an intact host, it will become intact also (or exposed if one of its neighbors is still compromised).

exposed. This rule is enforced immediately after any of the previous three rules are applied.

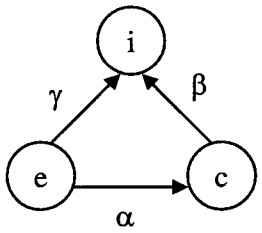

Figure 3: State transition without Rule 4. Here c, e, i means compromised, exposed, and intact respectively. At the end of each time step in the stochastic process, the states of the vertices are changed according to this state transition graph. Immediately after that, Rule 4 is applied to update the states of all the vertices.

α corresponds to attackers' capability (how fast hosts can be compromised); it is the probability that a node is successfully attacked during a time step (by a host compromise attack). β corresponds to defenders' capability; it is the probability of a successful proxy network reconfiguration during a time step. γ reflects how much memory attackers have and the coordination among attackers. For the most powerful attackers, which are fully coordinated and have infinite memory, they keep and share the information about the location of all the compromised and the exposed nodes; in this case, γ is equal to β. For less powerful attackers, which are memoryless and uncoordinated, for example, worms or automated attack tools [4, 5, 3, 9, 10], $\gamma = 1$. Normally, γ is between β and 1.

A few issues should be noted. First, we assume that there are always enough hosts in the resource pool to allow proxy network reconfiguration. This assumption has been validated in our previous work [26], which proved that when proactive reset mechanisms [3] are used, it is possible to keep the majority of hosts intact in a resource pool [4]. Second, in order to make analysis tractable, we assume uncorrelated host vulnerabilities (constant compromise probability α). In future work, correlation is clearly an important challenge to get a more complete understanding of the problem. However, as a first step to build the foundation, we assume uncorrelated vulnerabilities. How this assumption affects our results is discussed in section 3.3.4.

2.6 Problem Definition

In the model $M(G, \alpha, \beta, \gamma)$ described above, we define a graph G to be *robust* if almost surely all vertices of G can be changed into the intact state after a long run; we define a graph G to be *vulnerable* if with some non-trivial probability (greater than a positive constant) there always exist a significant number of compromised vertices in G at any time t.

Mathematically, we study the following problems in this paper:

1. *Given parameters α, β and γ, characterize the class of graphs G that are robust.*

2. *Given parameters α, β and γ, characterize the class of graphs G that are vulnerable.*

3. ANALYTICAL RESULTS

3.1 Theorems for Proxy Network Robustness

As discussed in Section 2.5, the system parameters α, β, γ characterize the capabilities of attackers and defenders. Intuitively, we also know that properties of the proxy network topology affect the impact of attacks. For example, attackers' impact can propagate quickly in a well connected graph, because once a node is compromised, many other nodes will be exposed. Theorems 1 and 2 precisely characterize the impact of these factors, showing how the system parameters α, β, γ and the spectra (or eigenvalues) of the proxy network topology graph G determine the robustness of the proxy network.

THEOREM 1. For the model $M(G, \alpha, \beta, \gamma)$, let σ_1 be the largest eigenvalue of the adjacency matrix of G, then the graph G is robust if $\frac{\beta(\alpha+\gamma)}{\alpha} > \sigma_1$. In particular, for any initial states, almost surely all compromised and exposed vertices vanish after $O(\frac{\alpha+\beta+\gamma-\alpha\sigma_1}{\alpha\beta+\beta\gamma-\alpha\sigma_1}\log n)$ steps. n is the number of vertices in G.

THEOREM 2. For the model $M(G, \alpha, \beta, \gamma)$, let $\bar{\lambda} = \max_{i\neq 0}\|1-\lambda_i\|$, where λ_i are the Laplacian spectrum of G. The graph G is vulnerable if $\frac{\beta}{\alpha} < \frac{1}{\bar{\lambda}^2} - 1$. In this case, with some positive constant probability, the volume of the compromised vertices will reach $\Theta(\bar{\lambda}^2 \mathrm{vol(G)})$ within $\Theta(\frac{\alpha-\beta\bar{\lambda}^2}{(\beta/(\alpha+\gamma)+1)\bar{\lambda}^2-\alpha}\log n)$ steps. n is the number of vertices in G. The volume of a vertex set S, vol(S), is defined to be the sum of degrees of the vertices in S, i.e., $\mathrm{vol(S)} = \sum_{v\in S} d_v$.

Intuitively, attackers expand their impact over the proxy network topology (by compromising or exposing proxy nodes), and the graph properties σ_1 and $\bar{\lambda}$ used in Theorem 1 and 2 characterize how fast attackers' expansion can be; in the mean time, defenders suppress attackers' expansion and remove attackers' impact in the proxy network (by recovering compromised or exposed nodes). Roughly speaking, the ratio $\frac{\beta}{\alpha}$ can be viewed as the ratio between the rate of proxy network reconfigurations and the rate of host compromise. In other words, defenders can recover compromised or exposed nodes $\frac{\beta}{\alpha}$ times faster than attackers can compromise or expose them. These theorems quantify how the proxy network topology affects the balance between the two competing forces, attackers and defenders.

Theorem 1 gives a sufficient condition of a proxy network being robust. The eigenvalue σ_1 is an important property of a graph, characterizing graph connectivity. Informally,

[3]Recoveries are done proactively, not triggered by intrusion detectors. Examples include periodic system reload and so on.

[4][26] showed that by combining practical intrusion detection based recoveries and proactive resets at a rate close to the compromise rate, the majority of the hosts can be kept intact

we can treat σ_1 as an average vertex degree of the graph. [5] Theorem 1 says when the defenders' capability overpowers the attackers' by a factor of σ_1 [6], defenders are quick enough to suppress the propagation of attackers impact, and the proxy network is robust. Attackers' impact will be quickly removed from the proxy network regardless of the initial state (even if attackers can have many compromised nodes to begin with).

Theorem 2 gives a sufficient condition of a proxy network being vulnerable. The Laplacian spectrum $\bar{\lambda}$ is another important property to characterize graph connectivity. $\bar{\lambda}$ can characterize how a set of vertices expands to its neighborhood. For any graph, $0 \leq \bar{\lambda} \leq 1$, and a smaller $\bar{\lambda}$ implies richer connectivity and better neighborhood expansion of the graph (small vertex sets have many vertices in the neighborhood). Extensive discussions about Laplacian spectrum can be found in [6]. Theorem 2 shows that when attackers can outrun defenders roughly by a factor of $\bar{\lambda}^2$, the proxy network is vulnerable. In this case, even if attackers only have one node exposed at the beginning, the number of compromised (and exposed) nodes will quickly grow, and defenders can never cleanly remove them. More importantly, this theorem applies to any subgraph of a proxy network topology. If this condition holds in a subgraph of a proxy network, attackers' impact will linger in that part of the network forever and cannot be cleanly removed.

3.2 Design Principles

When designing proxy networks, there are several issues to be considered, such as the effectiveness of location-hiding, performance of the proxy network and fault tolerance. We focus on location-hiding, and our theorems reveal the relation between some key properties of proxy network topologies and the effectiveness of location-hiding; they also specify the classes of favorable and unfavorable topologies for proxy networks. Design principles can be derived from these theorems. From Theorem 1, we know that graphs with high average vertex degrees require defenders to act significantly faster than attackers to make proxy networks robust. This posts a difficult requirement on defenders; therefore such topologies are unfavorable for proxy networks. From Theorem 2, we know that graphs with large clusters of tightly connected nodes tend to be vulnerable, because attackers' impact can linger inside those large clusters forever and cannot be easily removed. Therefore such topologies should be avoided in proxy network design. On the other hand, in proxy network design, one should look for topologies with relatively low average vertex degrees and balanced distribution of connectivity, because such topologies are easy to become robust and do not have obvious vulnerable subgraphs. These design principles can be summarized as follows.

1. Topologies with high average vertex degrees are in general unfavorable.
2. Topologies with large clusters of tightly connected nodes should be avoided.
3. Proxy network designers should look for topologies with reasonably low average vertex degree (depending on the capability of the defensive mechanisms available to them), and balanced distribution of connectivity in the graph.

[5] We have $d_{min} \leq \sigma_1 \leq d_{max}$ for any graph G. d_{min} and d_{max} are respectively the smallest and the largest vertex degree of the graph. In particular, $\sigma_1 = d$ for any d-regular graph.

[6] More precisely, $\frac{\beta}{\alpha}(\alpha + \gamma) > \sigma_1$. We know $\beta \leq \gamma \leq 1$ and $\alpha + \beta$ is a non-trivial constance(close to 1). Therefore $\frac{\beta}{\alpha}$ is the deciding factor of the left-hand side of the inequality.

Vertex Degree	σ_1	$\frac{1}{\bar{\lambda}^2} - 1$
3	3	≈ 0.75
5	5	≈ 1.25
7	7	≈ 1.75
9	9	≈ 2.25
11	11	≈ 2.75
13	13	≈ 3.25
15	15	≈ 3.75
17	17	≈ 4.25
19	19	≈ 4.75
21	21	≈ 5.25
23	23	≈ 5.75

Table 1: Topological Properties of Random Regular Graph

3.3 Case Study

Besides providing design principles, another important use of our theorems is to evaluate the goodness of a given topology class and determine whether it is appropriate for proxy networks. In this section, we present a case study to exemplify this use.

We study the following classes of candidate topologies for proxy networks: random regular graphs and current overlay networks (Chord [25] and CAN [21] as representatives). Random regular graphs are obvious candidates, because they are well understood and have many nice properties such as good connectivity. Existing overlay networks, such as Chord [25], CAN [21], Pastry [22] and Tapestry [29], are also reasonable candidates for proxy networks, and Chord has been proposed for location-hiding [17, 24]. However, it was not well understood whether those topologies are suitable for location-hiding. Therefore, it is worthwhile to evaluate such topologies.

Our previous work [26] showed that proxy networks with small diameters are not suitable for location-hiding. This result is applied along with Theorems 1 and 2 to all classes of graphs below.

3.3.1 Random Regular Graphs

A random d-regular graph is a regular graph with vertex degree d, in which connections between vertices are determined in some random way. We know that $\sigma_1 = d$ and $\frac{1}{\bar{\lambda}^2} - 1 \approx \frac{d}{4}$ for such graphs. Topological properties of some random regular graphs are shown in Table 1.

It is straightforward to observe that random regular graphs with high vertex degrees have large $\frac{1}{\bar{\lambda}^2} - 1 (\approx \frac{d}{4})$ and they are fairly easy to become vulnerable. Therefore they are not suitable for location-hiding. On the other hand, random regular graphs with low vertex degrees , have small σ_1

N	Diameter	σ_1	$\frac{1}{\lambda^2} - 1$
128	4	13	1.086
256	4	15	0.859
512	5	17	0.710
1024	5	19	0.604
2048	6	21	0.526
4096	6	23	0.465

Table 2: Topological Properties of Chord

and it is less demanding for defenders to make the proxy network robust. Therefore they are valid candidates for proxy networks.

3.3.2 Chord

Chord [25] topology is a regular graph with degree $2log_2N-1$, where N is the number of vertices in the graph. Consider a Chord network with $N = 2^m$ nodes, and each node is given a unique ID between 0 and $N-1$; there is an edge between vertices i and j if and only if $|i-j| = 2^k$, where $0 \leq k \leq (m-1)$ is an integer (Figure 4). Since Chord topology is a regular graph, we know that $\sigma_1 = 2log_2N - 1$. We also calculated the diameter and $\bar{\lambda}$ for such topologies. The results are summarized in Table 2.

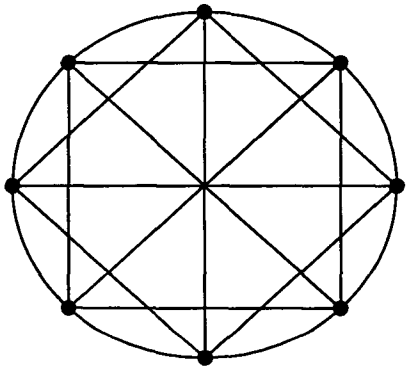

Figure 4: Chord Network Topology Graph with $N = 8$

It is straightforward to observe that Chord networks with reasonable diameters (≥ 5 for example) have large σ_1. This is an unfavorable property due to Theorem 1. We also studied Pastry [22] and Tapestry [29], and found that they have similar properties as Chord (In fact, they are a little worse than Chord, because they have larger σ_1 and larger $\frac{1}{\lambda^2} - 1$). We only present Chord as a representative case.

3.3.3 CAN

CAN [21] is less popular than the other three overlay networks. It uses a d-dimensional Cartesian space torus to construct the overlay network. It is a regular graph with degree $2d$. A CAN network using 2-dimensional torus, which has 9 nodes, is illustrated in Figure 5.

Because CAN networks are regular graphs, we know that $\sigma_1 = 2d$ (d is the dimension for the torus). Furthermore,

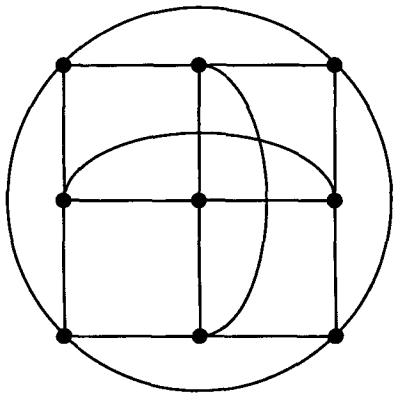

Figure 5: CAN Network Topology Graph with $N = 9, d = 2$

N	d	Diameter	σ_1	$\frac{1}{\lambda^2} - 1$
128	3	8	6	≈ 0
128	4	7	8	≈ 0
256	3	10	6	≈ 0
256	4	8	8	≈ 0
512	3	12	6	≈ 0
512	4	10	8	≈ 0
1024	3	16	6	≈ 0
1024	4	12	8	≈ 0
2048	3	19	6	≈ 0
2048	4	14	8	≈ 0
4096	3	24	6	≈ 0
4096	4	16	8	≈ 0

Table 3: Topological Properties of CAN

from the properties of d-dimensional toruses, we also know that the diameter of CAN networks is roughly $\sqrt[d]{N} \times \frac{d}{2}$ (N is the number of nodes in the network), and $\bar{\lambda} \approx 1$ (therefore $\frac{1}{\lambda^2} - 1 \approx 0$). The results for some CAN networks are shown in Table 3.

These results imply that the CAN network topology is more favorable for proxy networks than Chord (Pastry and Tapestry). With appropriate settings (choice of N and d), CAN networks can easily have large diameters and reasonably low σ_1, so that proxy networks can fairly easily become robust.

3.3.4 Discussion

Among the classes of topologies we studied, the CAN network and random regular graphs with low vertex degrees are favorable choices for proxy networks. Proxy networks with such topologies are easy to achieve robustness. On the other hand, popular overlays, such as Chord, Pastry and Tapestry, have fairly large σ_1, and therefore hard to achieve robustness. They are less favorable for proxy networks.

Chord, Pastry and Tapestry are designed to have rich connectivity to support efficient communication at the overlay level, a completely different goal. Therefore, it is not surprising that these topologies are not well suited for proxy

networks. On the other hand, CAN is also designed for similar purposes as Chord and other overlays, but there are two key differences that make CAN more favorable for proxy networks. First, topologically, CAN network have less connectivity than Chord and other overlays, and its neighborhood expansion property is not as good as Chord either. It is a merit for proxy networks, because attackers cannot quickly expand their impact. Second, CAN has a separate parameter (dimension of Cartesian space for the torus) to control the connectivity of the graph. This makes it flexible for us to choose a right amount of connectivity and an appropriate graph diameter to meet our requirement. In Chord and other overlays, however, graph connectivity is decided by the number of nodes in the network (for example, Chord has vertex degree $2log_2N - 1$); it is impossible to adjust the graph diameter and connectivity independently.

Our assumption that the compromise of distinct systems is uncorrelated makes analysis of the system dynamics tractable. The successful analysis provides significant insights into the relative benefits of different overlay network topologies. An interesting direction for future work is to extend our model to include correlation in compromise, but there are significant challenges in analysis. However, we conjecture that the result of a model with correlated host compromises might have a significant quantitative impact on the results but not a significant qualitative impact; that is it would not change the relative desirability of topologies. The reason for this is that correlation will exacerbate the negative impacts of high connectivity in topologies. Compared to our results, topologies vulnerable to attack will suffer even more greatly.

In this paper, we focus on the effectiveness of location-hiding. However, appropriate connectivity is also an important issue for proxy networks to achieve reasonable performance and fault tolerance. In fact, there is a fundamental trade-off between connectivity of proxy networks and the effectiveness of location-hiding they can provide, because richer connectivity implies more information stored on each proxy node, therefore attackers can gain information faster and propagate faster. In the design of proxy networks, we need to choose a right amount of connectivity to achieve both effective location-hiding and good performance and fault-tolerance. The value of our theorems is to provide a tool to quantify the appropriate level of connectivity for location-hiding, which is an important piece in proxy network design.

4. MATHEMATICAL PROOF

4.1 Basic facts on the spectra of graphs

Eigenvalues or the spectrum are very useful for controlling many graph properties. It have many applications including information retrieval [19], low rank approximation [1], and computer vision [13]. It has a rich history in the literatures (see [2, 6, 11, 8, 16, 15, 18, 27, 7]). The eigenvalues of many classes of graphs have been computed. For example, for random graph $G(n,p)$, the largest eigenvalue of its adjacency matrix is $(1+o(1))np$ while the rest of eigenvalues are bounded by $(1+o(1))2\sqrt{np}$, for $np = \Omega(1)$. The distribution of the eigenvalues of $G(n,p)$ follows Wigner's semi-circle Law. Recently, Chung, Lu and Vu [7] examined the eigenvalues of a random power law graph and proved that the Laplacian eigenvalues of the random power law graph also follows Wigner's semi-circle Law.

We will begin with some basic definitions. Let G be a connected (undirected) graph G. The adjacency matrix A of the graph G is defined as $A(x,y) = 1$ if x is adjacent to y, and 0 otherwise. The eigenvalues of A is denoted by $\sigma_1, \sigma_2, \ldots, \sigma_n$ in the decreasing order. Here σ_1 is the largest eigenvalue of G. For d-regular graph, σ_1 is just d. In general, $\sqrt{d_{\max}}\sigma_1 \leq d_{\max}$, where $d_{\max}$ denotes the maximum degree of G. We remark that the lower bound of σ_1 is achieved by a star of $d_{\max}+1$ vertices.

The *Laplacian eigenvalues* (or the *spectrum*) are also widely used in the spectral graph theory. It is defined as follows. Let d_v denote the degree of the vertex v, and T denote the diagonal matrix with (v,v)-th entry having value d_v. The *Laplacian* of G is defined to be the matrix

$$\mathcal{L} = I - T^{-1/2}AT^{-1/2}.$$

Here I is an identity $n \times n$ matrix. The Laplacian eigenvalues of G are defined as the eigenvalues of $\mathcal{L}$. They are often written in an increasing order:

$$\lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{n-1}.$$

For connected graph G, $\lambda_0 = 0$, $\lambda_1 > 0$, and $\lambda_{n-1} \leq 2$.

For example, the Laplacian eigenvalues of a cycle C_n are $1 - \cos\frac{2\pi k}{n}$ for $k = 0, \ldots, n-1$. The laplacian eigenvalues of a path P_n are $1 - \cos\frac{k\pi}{n-1}$ for $k = 0, \ldots, n-1$. Let G_1 and G_2 are two graphs of size n_1 and n_2. The cartesian product $G_1 \Box G_2$ of G_1 and G_1 is defined as a graph on $n_1 \times n_2$ vertices. The edges are added to the pair (u_1, u_2) and (v_1, v_2) if and only if $u_1 = v_1$ and $u_2v_2 \in E(G_2)$ or $u_2 = v_2$ and $u_1v_1 \in E(G_1)$. The spectrum of $G_1 \Box G_2$ can be computed as follows. We have

$$\lambda_1(G_1 \Box G_2) = \frac{1}{2}min\{\lambda_1(G_1), \lambda_1(G_2)\}$$

$$\lambda_{n_1n_2-1}(G_1 \Box G_2) = \frac{1}{2}(\lambda_{n_1-1}(G_1) + \lambda_{n_2-1}(G_2)).$$

In particular, for the d-dimensional Torus graph C_n^d, the $\bar{\lambda} = 1$ if n is even, $\bar{\lambda} = \max\{\cos\frac{\pi}{n}, 1 - \frac{1}{d} + \frac{1}{d}\cos\frac{2\pi}{n}\}$ if n is odd.

If G is d-regular graph, the Laplacian becomes $\mathcal{L} = I - \frac{1}{d}A$. Thus, $\lambda_i = 1 - \frac{1}{d}\sigma_{n-i}$ for $0 \leq i \leq n-1$. In general, the spectrum of the graph G can be very different from the eigenvalues of the adjacency matrix.

Laplacian eigenvalues control the expansion rate of the neighborhoods for any subset S. We have the following useful lemma.

Lemma (see [6]): *Suppose G is not a complete graph. For $S \subset V(G)$, the neighborhood $N(S)$ satisfies*

$$\frac{\text{volN(S)}}{\text{vol(S)}} > \frac{1}{\bar{\lambda}^2 + (1-\bar{\lambda}^2)\frac{\text{vol(S)}}{\text{vol(G)}}},$$

where $\bar{\lambda} = \max_{i \neq 0} \|1 - \lambda_i\|$.

4.2 Proof of Theorem 1

Let f_v^t (or g_v^t) be the probability that the node v is compromised (or exposed) at time t respectively. We have the following recurrence formula:

$$\begin{cases} f_v^{t+1} = (1-\beta)f_v^t + \alpha g_v^t \\ g_v^{t+1} = (1 - f_v^{t+1})(1 - \prod_{u \sim v}(1 - f_u^{t+1})) \\ \quad + g_v^t(1 - \alpha - \gamma)\prod_{u \sim v}(1 - f_u^{t+1}), \end{cases}$$

for every vertex v and time t. Here $u \sim v$ means uv is an edge. The first additive item in g_v^{t+1} is the contribution due to a neighbor of v is captured at time $t+1$. The second item is the probability that a vertex being exposed at time t and remains being exposed at $t+1$. We can rewrite is as

$$\begin{cases} f_v^{t+1} = (1-\beta)f_v^t + \alpha g_v^t \\ g_v^{t+1} = (1-(1-\beta)f_v^t - (1-\gamma)g_v^t)(1-\prod_{u\sim v}(1-f_u^{t+1})) \\ +g_v^t(1-\alpha-\gamma). \end{cases}$$

The above recurrence formula is not easy to solve. However, we have

$$g_v^{t+1} \leq \sum_{u\sim v} f_u^{t+1} + g_v^t(1-\alpha-\gamma).$$

Here we use the inequality $(1-\prod_{u\sim v}(1-f_u^{t+1})) \leq \sum_{u\sim v} f_u^{t+1}$. Let f^t be the column vector with i-th entry f_i^t. Let g^t be the column vector with i-th entry g_i^t. We have

$$\begin{aligned} f^{t+1} &= (1-\beta)f^t + \alpha g^t \\ g^{t+1} &\leq Af^{t+1} + (1-\alpha-\gamma)g^t, \end{aligned}$$

where A is the adjacency matrix of G. Given two vectors X and Y, the notation $X \leq Y$ means $X_i \leq Y_i$ for every index i. We can rewrite into the following matrix form.

$$\begin{pmatrix} I & \\ -A & I \end{pmatrix}\begin{pmatrix} f^{t+1} \\ g^{t+1} \end{pmatrix} \leq \begin{pmatrix} (1-\beta)I & \alpha I \\ & (1-\alpha-\gamma)I \end{pmatrix}\begin{pmatrix} f^t \\ g^t \end{pmatrix}.$$

We (left) multiply both hand sides by a non-negative matrix

$$\begin{pmatrix} I & \\ A & I \end{pmatrix}.$$

We have

$$\begin{pmatrix} f^{t+1} \\ g^{t+1} \end{pmatrix} \leq \begin{pmatrix} (1-\beta)I & \alpha I \\ (1-\beta)A & \alpha A + (1-\alpha-\gamma)I \end{pmatrix}\begin{pmatrix} f^t \\ g^t \end{pmatrix}.$$

Let M denote the square matrix in the above inequality. We have

$$\begin{aligned} &\lambda I - M = \begin{pmatrix} (\lambda-1+\beta)I & -\alpha I \\ -(1-\beta)A & -\alpha A + (\lambda-1+\alpha+\gamma)I \end{pmatrix} \\ &= B\begin{pmatrix} (\lambda-1+\beta)I & \\ & (\lambda-1+\alpha+\gamma)I - \frac{\lambda\alpha}{\lambda-1+\beta}A \end{pmatrix}B^{-1}, \\ &where B = \begin{pmatrix} I & \\ -\frac{1-\beta}{\lambda-1+\beta}A & I \end{pmatrix}. \end{aligned}$$

Thus, we obtain

$$\det(\lambda I - M) = \prod_{i=1}^{n}((\lambda-1+\beta)(\lambda-1+\alpha+\gamma) - \lambda\alpha\sigma_i).$$

Here $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ are the eigenvalues of A. The largest eigenvalue μ_1 of M satisfies the equation

$$f(x) = (x-1+\beta)(x-1+\alpha+\gamma) - \alpha\sigma_1 x = 0.$$

It is less than 1 if and only if $f(1) = \beta(\alpha+\gamma) - \alpha\sigma_1 > 0$. By solve $f(x) = 0$, we have the following upper bound:

$$\mu_1 \leq \frac{\alpha+\beta+\gamma-\alpha\beta-\beta\gamma}{\alpha+\beta+\gamma-\alpha\sigma_1} < 1.$$

For any initial state, almost surely there is no compromised or exposed vertex after $O(\frac{\log n}{-\log \mu_1}) = O(\frac{\alpha+\beta+\gamma-\alpha\sigma_1}{\alpha\beta+\beta\gamma-\alpha\sigma_1}\log n)$ steps. □

4.3 Proof of Theorem 2

Let S_t (or T_t) be the set of compromised (or exposed) nodes at time t respectively. Let X_t be the volume of the set of compromised nodes, i.e., $X_t = \mathrm{vol}(\mathrm{S_t})$. Let $Y_t = \mathrm{vol}(\mathrm{T_t})$ be the volume of the set of exposed nodes. We have

$$\begin{aligned} E(X_{t+1}) &= (1-\beta)E(X_t) + \alpha E(Y_t) \\ E(Y_{t+1}) &= (1-\alpha-\gamma)E(\mathrm{vol}(\mathrm{T_t} \setminus (\mathrm{N}(\mathrm{S_{t+1}}) \setminus \mathrm{S_{t+1}}))) \\ &\quad + E(\mathrm{vol}(\mathrm{N}(\mathrm{S_{t+1}}) \setminus \mathrm{S_{t+1}})) \\ &\geq (1-\alpha-\gamma)E(Y_t) + (\alpha+\gamma)E(\mathrm{vol}(\mathrm{N}(\mathrm{S_{t+1}}) \setminus \mathrm{S_{t+1}})) \end{aligned}$$

From Lemma 4.1, for at subset S with $\mathrm{vol(S)} \leq \epsilon\mathrm{vol(G)}$, we have

$$\mathrm{vol(N(S))} > \frac{\mathrm{vol(S)}}{\bar{\lambda}^2 + (1-\bar{\lambda}^2)\epsilon}.$$

Let $\delta = \frac{1}{\bar{\lambda}^2+(1-\bar{\lambda}^2)\epsilon} - 1$. The following recurrence formula holds as long as $S_{t+1} \leq \epsilon\mathrm{vol(G)}$.

$$\begin{aligned} E(X_{t+1}) &= (1-\beta)E(X_t) + \alpha E(Y_t) \\ E(Y_{t+1}) &\geq (1-\alpha-\gamma)E(Y_t) + (\alpha+\gamma)\delta E(X_{t+1}) \end{aligned}$$

We can rewrite it into the following form.

$$\begin{aligned} &\begin{pmatrix} 1 & \\ -(\alpha+\gamma)\delta & 1 \end{pmatrix}\begin{pmatrix} E(X_{t+1}) \\ E(Y_{t+1}) \end{pmatrix} \\ &\geq \begin{pmatrix} 1-\beta & \alpha \\ & 1-\alpha-\gamma \end{pmatrix}\begin{pmatrix} E(X_t) \\ E(Y_t) \end{pmatrix}. \end{aligned}$$

We left-multiply the both hand sides by a non-negative matrix

$$\begin{pmatrix} 1 & \\ (\alpha+\gamma)\delta & 1 \end{pmatrix}.$$

We have

$$\begin{pmatrix} E(X_{t+1}) \\ E(Y_{t+1}) \end{pmatrix} \geq \mathrm{M}\begin{pmatrix} E(X_t) \\ E(Y_t) \end{pmatrix},$$

where $M = \begin{pmatrix} 1-\beta & \alpha \\ (\alpha+\gamma)\delta(1-\beta) & (1-\alpha-\gamma)+\alpha(\alpha+\gamma)\delta \end{pmatrix}$. The characteristic polynomial $p(x)$ of M is

$$p(x) = (x-1+\beta)(x-1+\alpha+\gamma) - \alpha(\alpha+\gamma)\delta x.$$

Since $p(1) = (\beta-\alpha\delta)(\alpha+\gamma)$, the largest eigenvalue $\sigma(M)$ of M is greater than 1 if $\beta < \alpha\delta$. In this case, we have

$$\sigma(M) \geq \frac{\alpha+\beta+\gamma-\alpha\beta-\beta\gamma}{\alpha+\beta+\gamma-\alpha(\alpha+\gamma)\delta}.$$

Let (c_1, c_2) be the corresponding eigenvector of $\sigma(M)$ so that $(c_1, c_2)M = \sigma(M)(c_1, c_2)$. Then, both c_1 and c_2 are positive. The expect value of $c_1X_t + c_2Y_t$ increases by a factor of at least $\sigma(M) > 1 + \frac{(\alpha\delta-\beta)(\alpha+\gamma)}{\alpha+\beta+\gamma-\alpha^2\delta-\alpha\gamma\delta}$ until $X_t \geq \epsilon\mathrm{vol(G)}$.

Let $Z_t = c_1X_t + c_2Y_t$. The above statement shows the expected value of Z_t grows exponentially (as a function of t). By the recurrence formula of $E(X_t)$ and $E(Y_t)$, both expected values of X_t and Y_t will grow exponentially. It is sufficient to show Z_t grow exponentially with constant probability.

By Chernoff's Inequality, we can show Z_t concentrates on its expected value. There exists a absolute constant c so the following statements holds.

$$Pr(Z_t > (1-\varepsilon)E(Z_t)) \leq e^{-c\varepsilon^2 E(Z_t)}.$$

Since $E(Z_t)$ increases by a factor of $\sigma(M)$, $\sum_{t\geq 0} e^{-c\epsilon^2\sigma^t(M)}$ converges and there exists an absolute constant t_0 that $\sum_{t\geq t_0} e^{-c\epsilon^2 E(Z_t)} < \frac{1}{2}$. Moreover, there is a constant probability that $Z_t \geq E(Z_{t_0})$ for some $t \leq 2t_0$. Hence, with positive constant probability, Z_t will grow at least by a factor $\frac{1+\sigma(M)}{2} > 1$ until X_t reach ϵvol(G).

We choose $\epsilon = O(\bar{\lambda}^2)$ so that $\delta \approx \frac{1}{\lambda^2} - 1$. With constant probability, Z_t, X_t, Y_t will reach $\Theta(\bar{\lambda}^2 \text{vol(G)})$ within $\Theta(\frac{(\alpha\delta-\beta)(\alpha+\gamma)}{\alpha+\beta+\gamma-\alpha^2\delta-\alpha\gamma\delta}\log n) = \Theta(\frac{\alpha-\beta\bar{\lambda}^2}{(\beta/(\alpha+\gamma)+1)\bar{\lambda}^2-\alpha}\log n)$ steps. The proof of theorem 2 is finished. □

5. RELATED WORK

The most related work is our previous work [26], which studied the impact of proxy network reconfiguration mechanisms, and proved that with reconfiguration mechanisms, proxy networks can effectively hide applications' IP addresses thereby preventing infrastructure-level DoS attacks. [26] also proved the importance of proxy network topology, but did not study how to choose a topology.

The differences between [26] and this work are the following. [26] studied the possibility of using overlay networks to achieve location-hiding, and it focused on the impact of defensive mechanisms, such as resource recovery and proxy network reconfiguration mechanisms. It assumed having a favorable topology for defenders, and did not address how to design proxy network topologies. This work, on the other hand, studies the impact of proxy network topologies on location-hiding, and focuses on how to design proxy network topologies to effectively achieve location-hiding. They are complementary research.

There are other researchers exploring the use of overlay network to tolerate or avoid DoS attacks. The Secure Overlay Services (SOS)[17] uses Chord[25] as the overlay network to provide some amount of anonymity to hide the location of secret "servlets". There are primitive analysis about the system security under simple attack models such as DoS attack on individual hosts. However, the analysis is tied to the Chord-based design, and cannot be generalized. Furthermore, they did not consider host compromise attacks, which are critical threats to their scheme. Internet Indirection Infrastructure (i3) [24] also suggested the use of Chord for location-hiding. But they did not fully analyze the effectiveness of their scheme and did not consider host compromise attacks either.

6. SUMMARY AND FUTURE WORK

In this paper, we studied how the topological properties of proxy networks affect their effectiveness to achieve location-hiding. In particular, we presented two theorems to characterize when proxy networks are robust against attacks (attackers' impact can be quickly and completely removed from the proxy network), and when proxy networks are vulnerable to attacks (attackers' impact can linger forever and never be completely removed). We applied these theorems to a set of topologies. From our results, we found that Chord [25] topology, which has been suggested for location-hiding [17, 24], is in fact not a favorable topology for such purposes. Our results also showed that CAN [21], a less popular overlay network, can be a favorable topology for location-hiding.

Our theoretical results and case study lead to a few design principles: proxy networks with high average vertex degrees are in general unfavorable, because it is hard to make them robust; furthermore, proxy networks should avoid large clusters of tightly connected nodes, because such clusters are vulnerable to attacks. On the other hand, graphs with low average vertex degrees and balanced distribution of connectivity are in general good candidates for proxy networks; because they are reasonably easy to become robust and they do not have vulnerable regions to harbor attackers' impact.

Our theorems also provide a tool to evaluate the goodness of a proxy network topology for location-hiding. In the design of proxy networks, it can help us select favorable topologies from a set of candidates and filter out undesirable ones.

Our future work have the following directions. On the theoretical path, we will extend our model to include correlated host vulnerabilities, and study the impact of such correlations, and the impact of technologies such as intrusion detection, intrusion/fault containment and software heterogeneity. We will also study how to design proxy networks that can tolerate massive proxy failures due to DoS attacks. The key challenges include finding appropriate system reconfiguration schemes and appropriate proxy network topologies with sufficient connectivity to tolerate proxy failures. Furthermore, we can generalize our framework and apply our work to wider classes of problems. For instance, our work can be extended to study the spreading of computer viruses or even problems outside the computer science domain. On the empirical path, we will implement a prototype proxy network based on our theoretical results and empirically verify the correctness of our theoretical results.

7. ACKNOWLEDGMENTS

Supported in part by the Defense Advanced Research Projects Administration through United States Air Force Rome Laboratory Contracts AFRL F30602-99-1-0534 and the National Science Foundation through NSF EIA-99-75020 Grads and NSF Cooperative Agreement ANI-0225642 (OptIPuter)to the University of California, San Diego, and NSF Grants DMS 0100472 and ITR 0205061. Support from Hewlett-Packard is gratefully acknowledged.

8. REFERENCES

[1] D. Achlioptas and F. McSherry. Fast computation of low-ranked matrix approximation. *STOC*, pages 611–618, 2001.

[2] N. Biggs. *Algebraic Graph Theory.* Cambridge University Press, 1993.

[3] CERT. "code red ii:" another worm exploiting buffer overflow in iis indexing service dll, 2001.

[4] CERT. "code red" worm exploiting buffer overflow in iis indexing service dll, 2001.

[5] CERT. Cert advisory ca-2003-04 ms-sql server worm, 2003.

[6] F. Chung. *Spectral Graph Theory.* AMS Publications, 1997.

[7] F. Chung, L. Lu, and V. Vu. The spectra of random graphs with given expected degrees. *Proceedings of National Academy of Science*, 100:6313–6318, 2003.

[8] A. Crisanti, G. Paladin, and A. Vulpiani. *Products of Random Matrices in Statistical Physics*. Springer Series in Solid-State Sciences. Springer, Berlin, 1993.

[9] D. Dittrich. The dos project's "trinoo" distributed denial of service attack tool, 1999.

[10] D. Dittrich. The "tribe flood network" distributed denial of service attack tool, 1999.

[11] P. ErdHos and T. Gallai. Graphs with points of prescribed degrees. *Mat. Lapok*, 11:264–274, 1961.

[12] B. Fonseca. Yahoo outage raises web concerns, 2000.

[13] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral partitioning with indefinite kernels using the nystrom extension. In *European Conference on Computer Vision*, Copenhagen, 2002.

[14] D. Frank. Cybersecurity called key to homeland defense, 2001.

[15] Z. Furedi and J. Komlos. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3):233–241, 1981.

[16] K.-I. Goh, B. Kahng, and D. Kim. Spectra and eigenvectors of scale-free networks. *Phy. Rev. E*, 64(051903), 2001.

[17] A. D. Keromytis, V. Misra, and D. Rubenstein. Sos: Secure overlay services. In *ACM SIGCOMM'02*, Pittsburgh, PA, 2002. ACM.

[18] F. Kirchhoff. Über die auflösung der gleichungen, auf welche man bei der untersuchung der linearen verteilung galvanischer ströme geführt wird. *Ann. Phys. chem.*, 72:497–508, 1847.

[19] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms*, 1999.

[20] J. Miller. 2004 it budget request focuses on homeland defense, cybersecurity, 01/20/2003 2003.

[21] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *ACM SIGCOMM'01*, 2001.

[22] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001*, 2001.

[23] F. B. Schneider. *Trust in Cyberspace*. National Academy Press, 1999.

[24] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet indirection infrastructure. In *SIGCOMM*, Pittsburge, Pennsylvania USA, 2002.

[25] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *ACM SIGCOMM'01*, pages 149–160, 2001.

[26] J. Wang and A. A. Chien. Using overlay networks to resist denial-of-service attacks. Technical report, CSE Department, University of California, San Diego, 2003.

[27] E. P. Wigner. On the distribution of the roots of certain symmetric matrices. *The Annals of Mathematics*, 67:325–327, 1958.

[28] M. Williams. Ebay, amazon, buy.com hit by attacks, 2000.

[29] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz. Tapestry: A resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*, 2003.