APPROVAL SHEET

Title of Thesis: Spread Identity: A New Dynamic Address Remapping Mechanism for Anonymity and DDoS Defense

Name of Candidate: Bhushan Ekanth Sonawane
Master of Science, 2010

Thesis and Abstract Approved: _____

Dhananjay Phatak
Associate Professor
Department of Computer Science and
Electrical Engineering

_____

Alan T. Sherman
Associate Professor
Department of Computer Science and
Electrical Engineering

Date Approved: _____

Curriculum Vitae

Name: Bhushan Eknath Sonawane.

Permanent Address: 1114, Courtney road, Baltimore, MD-21227.

Degree and date to be conferred: MS in Computer Science, August 2010.

Date of Birth: May 9, 1985.

Place of Birth: Dhule, Maharashtra, India.

Secondary education: Jai Hind Junior College, Dhule, Maharashtra.

Collegiate institutions attended:

University of Maryland Baltimore County, MS Computer Science, 2010.
University of Pune, BE Computer Engineering, 2006.

Major: Computer Science.

Professional Publications:

Alan T. Sherman, Dhananjay Phatak, Bhushan Sonawane, Vivek Relan, "Location Authentication through Power Line Communication: Design, Protocol, and Analysis of a New Out-of-Band Strategy" in Proceedings of the 14th IEEE International Symposium on Power Line Communications and its Applications, March 2010.

Professional positions held:

July 2006 — July 2008. Senior Member of Technical Staff, Airtight Networks, Pune.

ABSTRACT

Title of Thesis: Spread Identity: A New Dynamic Address Remapping Mechanism for Anonymity and DDoS Defense.

Bhushan Eknath Sonawane, MS-Computer Science, 2010.

Directed By: Dhananjay Phatak, Associate Professor, Department of Computer Science and Electrical Engineering.

Alan T. Sherman, Associate Professor, Department of Computer Science and Electrical Engineering.

We present and experimentally evaluate spread identity — a new dynamic network address remapping mechanism for Internet connections that provides anonymity and DDoS defense. For each session between a source and destination host, the trusted source gateway dynamically and randomly assigns an IP address for the source from the pool of all routable IP addresses within the source organization. Similarly, in response to a name resolution query from the source gateway, the trusted authoritative DNS server for the destination host dynamically and randomly assigns an IP address for the destination from the pool of all routable IP addresses within the destination organization. Moreover, different hosts can share the same IP address when communicating with distinct peers. Each gateway creates a NAT entry, valid for the communication session, based on the dynamic assignment by its organization. An eavesdropper listening to packets flowing through the Internet between the source

and destination gateways learns only the source and destination domains; the eavesdropper cannot see the actual complete IP addresses of the source and destination hosts. In addition, spread identity enhances DDoS defense capabilities by facilitating filtering of packets based on destination address. Whereas a traditional IP source address can be spoofed, with spread identity the destination address cannot be spoofed. Therefore, using multiple IP addresses for the same destination enables simple and powerful DDoS protections that block attackers without necessarily blocking legitimate users. Our ns-2 simulations demonstrate that file transfer success rates for our spread identity DDoS protection mechanism are similar to those of filter- and capability-based approaches, with lower file transfer times than for filter-based approaches. Deploying spread identity requires changes to organizational gateways but not to Internet routers. Another cost is increased DNS traffic, but unlike overlay-based DDoS defense approaches, spread identity does not increase overall communication network latency. A partial form of spread identity implemented only at the destination facilitates destination-based filtering without providing sender anonymity.

Spread Identity: A New Dynamic Address Remapping Mechanism for Anonymity and DDoS Defense.

By

Bhushan Eknath Sonawane

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County, in partial fulfillment
of the requirements for the degree of
MS Computer Science
2010

Dedicated to my family.

Acknowledgements

I would like to express my sincere gratitude to my advisor Dr. Dhananjay Phatak. I thank him for the constant support and guidance, and for his continued belief in me throughout this thesis work. I also want to thank my co-adviser, Dr. Alan T. Sherman, for all his valuable suggestions. During the course of my research, he has been a great source of inspiration, motivation, and support. I would like to thank Dr. Chintan Patel for graciously agreeing to be on my thesis committee.

I want to thank Vivek Relan for all the help he has given me during this thesis work. Thanks to my parents for their never-ending endurance with me during the tough times. I am grateful to my friends Varish, Tejas, and Tushar for being supportive throughout my stay at UMBC.

Table of Contents

List of Figures

Chapter 1

INTRODUCTION

Static mappings of identities create egregious vulnerabilities that undermine privacy and facilitate identity theft. For example, static passwords, static credit card numbers, static RFID tags, and static IP addresses simplify the tasks of a malicious adversary who wishes to track individuals or misuse their credentials. By contrast, dynamic passwords [1], dynamic credit card numbers [2], dynamic RFID tags (*e.g.,* as realized by hash chains), and dynamic *Internet Protocol (IP)* addresses can greatly complicate the adversary's job. In this paper, we propose and analyze a new and powerful approach for dynamically "spreading the identities" (remapping the IP addresses) of source and destination hosts among a pool of identities (respectively, within a source and destination organization) for Internet sessions. This approach, which we call *spread identity,* enhances network anonymity and enables new effective responses to many *Distributed Denial of Service (DDoS)* attacks. Spread identity also provides a limited traceback capability.

In common practice today, when a client establishes a session with a server over the Internet, the connection is established using a static mapping of IP addresses and host names. Consequently, an eavesdropper can discover which client is communicating with which server, violating the communicants' privacy. Furthermore, a malicious adversary can easily concentrate a DDoS attack against a specific server by directing her evil accomplices to target a particular static IP address. We show how spread identity applied to Internet communications (henceforth referred to simply as spread identity) provides a solution to these two problems.

We describe how to implement spread identity at an organizational level (*e.g.*, for a domain such as www.umbc.edu) to support inter-organizational communications. Each organization has a gateway which performs dynamic address translations valid for one session. For example, the sender's organizational gateway spreads the identity of each of its clients among its pool of routable IP addresses assigned to it by its *Internet Service Provider (ISP).* Similarly, the destination's organizational gateway spreads the identity of each of its hosts among its pool of routable IP addresses. At the source gateway, the translation is performed by gateway routers using Network *Address Translation (NATing).* At the destination, the translation is performed by the authoritative DNS server in cooperation with the destination gateway.

Phatak [3] first proposed the concept of spread identity in 2005. We extend and improve this initial work by pooling all routable IP address at each organization gateway and by reusing IP addresses through controlled NATing. These improvements mitigate restrictions in the original concept caused by a limited set of routable IP addresses under IPv4.

As a defensive against DDoS attacks, spread identity offers advantages over the three main existing defenses. In comparison with filters and capabilities, spread identity requires fewer changes to current infrastructure: spread identity requires changes only to organizational gateways, not to all network routers. Overlay strategies also require minimal infrastructure changes but are potentially vulnerable to compromise or bypassing of the servlets (*e.g.*, an attacker might learn the true IP address of the destination and send traffic directly there). Using the ns-2 network

simulator [4], we demonstrate the effectiveness of spread identity as a tool for responding to DDoS attacks.

Using the ns-2 simulator, we measured file transfer success ratios and file transfer times for different numbers of DDoS attacker and for various bottleneck link capacities. Our results demonstrate that the file transfer success ratios for our spread identity DDoS protection mechanism are similar to those of filter- and capability-based approaches, with lower file transfer times than for filter-based approaches.

It is possible to implement spread identity at the destination gateway only. Such a partial implementation still provides DDoS defensive capabilities and limited traceback capability but fails to protect sender anonymity.

Contributions of this paper include: (1) System architecture of spread identity for Internet communications. (2) Analysis of DDoS capabilities facilitated by spread identity. (3) Analysis of network-level anonymity effected by spread identity. (4) Experimental demonstration and evaluation of spread identity's DDoS defense capabilities.

The rest of the paper is organized as follows. Section 2 overviews the spread identity concept. Section 3 explains our spread identity architecture for Internet communications, including our assumptions, protocols, and design and performance issues. Section 4 reviews related work. Section 5 describes advantages of spread identity and gives a detailed system design for DDoS defense and anonymity. Section 6 presents our experimental results using the ns-2 simulator. Section 7 explains implementation challenges, open problems, and backward compatibility issues. Finally, Section 8 summarizes our conclusions. Appendix A lists acronyms and

abbreviations used in this paper. We assume the reader is familiar with the basics of computer network security, as presented by Kaufman, Perlman, and Speciner [5], for example.

Chapter 2

OVERVIEW OF SPREAD IDENTITY

In this section we give a brief overview of spread identity, focusing on its architecture, dynamic address remapping mechanisms at the source and destination organization gateways, benefits of network communication anonymity and DDoS defense, and engineering challenges to its deployment.

As shown in Figure 1, we assume a model in which a client in some organization *X* wishes to communicate with a server in another organization *Y*. A trusted gateway for organization *X* dynamically assigns an IP address to the source client, from among the pool of all routable IP addresses within *X*. Similarly, a trusted DNS server for organization *Y* dynamically assigns an IP address to the destination server, from among the pool of all routable IP addresses within *Y*. These translations are used only for the duration of the communication session. Unlike traditional NATing in which source IP addresses are assigned in a predictable way, spread identity assigns source IP addresses in a random, unpredictable way. Unlike traditional static DNS assignments of hostnames to destination IP addresses, spread identity dynamically assigns hostnames to destination IP addresses in a random, unpredictable way.

For simplicity, Figure 1 depicts organization *X* only as a source organization, and organization *Y* only as a destination organization. Typically, however, each organization would implement both source and destination spread identity. It is also possible to implement spread identity at the destination gateway only, albeit doing so would not reap its anonymity benefits.

Spread identity works by dynamically translating addresses at the source gateway and destination DNS server. At the source gateway, NAT table entries are created to enable the gateway to route packets to particular hosts and to filter packets without NAT table entries. The destination organization has an authoritative DNS server, which maps host names to IP addresses for that organization. With spread identity, each session must be preceded by a DNS query during which the source client discovers an IP address for the destination host, as dynamically assigned by the destination's authoritative DNS server. This DNS server stores triples of source IP address, destination host name, and dynamically assigned IP address of destination host. These dynamic NAT associations allow the same IP address to be assigned simultaneously to distinct destination hosts for distinct sources. Likewise, the same source IP address can be used simultaneously to support connections from distinct sources to distinct destinations. Furthermore, since incoming and outgoing traffic are handled separately, the same routable IP address can simultaneously support multiple incoming and multiple outgoing connections, subject to a few distinctness constraints. The DNS server also communicates its mappings of hostnames to IP addresses to its organizational gateway, to enable routing and filtering. We assume all DNS requests and responses are encrypted at the organizational gateways.

Although Figure 1 depicts only one gateway per organization, typically each organization would deploy multiple gateways for increased availability. Similarly, to avoid single point failures of network links, we assume organizations deploy multi-homing [6] with multiple links and multiple IP addresses. All communications in or out of any organization must travel through its gateways. Furthermore, each gateway

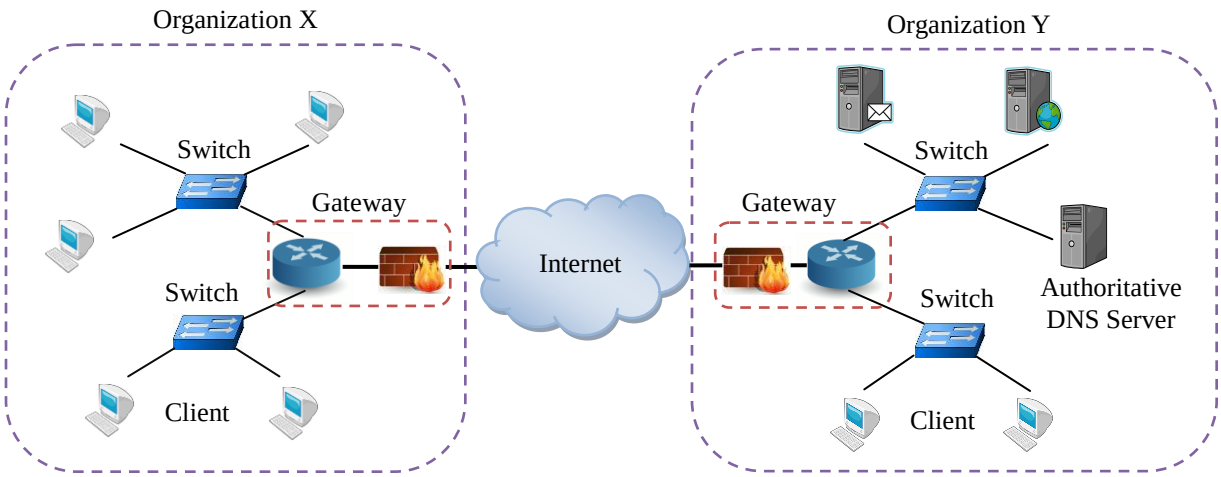

Figure 1: System architecture of spread identity for Internet communications. When a client in organization *X* initiates a communication session with a server in organization *Y*, the gateway for *X* dynamically and randomly assigns an IP address for the client from the pool of all routable IP addresses within *X*. In response to a name resolution query from *X*'s gateway, the authoritative DNS server for *Y* dynamically and randomly assigns an IP address for the destination server from the pool of all routable IP addresses within *Y*. Each gateway creates a NAT entry, valid for the communication session, based on the dynamic assignment by its organization.

includes a firewall, which hides the organization's private network from the external world.

Figure 2 shows how the source and destination gateways map IP addresses to hosts within their organizations. For example, for each session, the source gateway dynamically assigns to the source host an IP address randomly selected from the pool of all routable IP addresses within the source organization. Moreover, by considering transport layer port numbers, different hosts can share the same IP address when communicating with different destinations. Furthermore, different IP addresses can be assigned to the same host when communicating with different parties.

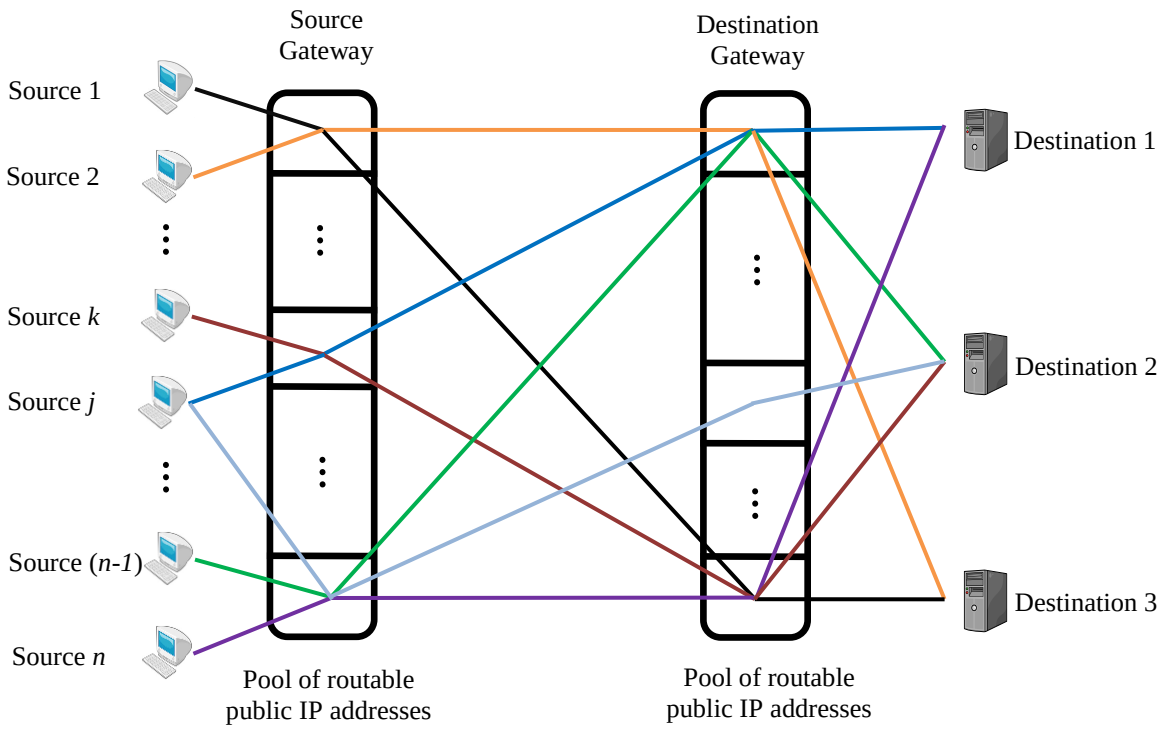

Figure 2: Network address translation tables of the source and destination gateways using spread identity. For each session, each gateway dynamically and randomly assigns to its communicant host an IP address from the pool of all routable IP addresses within its organization. Moreover, different hosts can share the same IP address when communicating with distinct peers. For example, in this figure, Source 1 and Source 2 share the same IP address. These dynamic address translations, augmented by address multiplexing, provide anonymity against network eavesdroppers. In addition, using multiple IP addresses for the same destination enables simple and powerful DDoS protections that block attackers without necessarily blocking legitimate users.

Spread identity for Internet communications achieves two major benefits: network-level anonymity and enhanced DDoS defense capabilities. Network-level anonymity is achieved through pseudonyms assigned by a trusted gateway at the source organization and by a trusted DNS server at the destination organization; as such, spread identity is similar to a one-layer mixnet [7]. For each session between a source host and a destination host, the source gateway dynamically assigns a

temporary pseudonym to the source host chosen as one of the routable IP addresses assigned to the source organization. Similarly, the destination DNS server assigns a temporary pseudonym to the destination host chosen as one of the routable IP addresses assigned to the destination organization. Thus, an eavesdropper listening to packets flowing through the Internet between the source and destination gateways learns only the source and destination domains; the eavesdropper cannot see the actual complete IP addresses of the source and destination hosts. Although the eavesdropper can link source and destination packets within any session, she cannot link packets between different sessions. Encrypting DNS requests and responses at gateways protects the established bindings of pseudonyms from the Internet eavesdropper.

When spread identity is implemented at both the source and destination organizations, the following enhanced variation is possible. Using a method analogous to spread spectrum radio broadcasts, the source and destination gateways could frequently change their address translations within a session, following a cryptographically-secure pseudorandom pattern derived from a shared secret key.

Spread identity enhances DDoS defense capabilities by facilitating filtering of packets based on destination address. Whereas a traditional IP source address can be spoofed, with spread identity the destination address cannot be spoofed. Furthermore, the destination address dynamically assigned for each source host serves as dynamic "flow marker". Therefore, filtering on destination address is easier and more effective than filtering on source addresses.

With spread identity, DDoS defense by destination filtering limits adverse impact on legitimate traffic. A typical DDoS attack attempts to clog the network bandwidth of a particular target IP address with traffic emanating from many evil hosts. Without spread identity, blocking all packets sent to the target would block both malicious and legitimate traffic. With spread identity, however, each session with the target uses a different destination IP address. Therefore, packets with suspicious destination addresses (*e.g.*, destination addresses that occur too frequently) can be filtered without blocking legitimate packets sent to the target from other sessions. Furthermore, with the cooperation of other gateways and routers, such destination filtering can be carried out inexpensively and closer to the source, lessening collateral bandwidth consumption from the attack.

Because the source host must learn the dynamically assigned destination IP address, spread identity facilitates a limited traceback capability (*i.e.*, ability to learn the origin of a packet). A DDoS attacker must first learn the address of her intended target. Therefore, with destination spread identity alone, the destination gateway knows the IP address of the machine to which it sent the requested destination address. While possibly incomplete, this IP address is useful even if the attacker shares the target address with her conspirators and even if this IP address is the last hop in an anonymizing network such as TOR.

Implementing spread identity raises several engineering challenges: orchestrating DNS caching at hierarchical DNS servers and host machines, handling loads on DNS servers, scaling gateways for larger organizations, and performing reverse DNS lookups. In the next section we propose solutions to these challenges.

Chapter 3

SPREAD IDENTITY ARCHITECTURE FOR INTERNET COMMUNICATIONS

We describe our spread identity architecture for Internet communications in terms of its components, assumptions, protocols, and design and performance issues.

3.1 Components

As shown in Figure 3, spread identity is implemented by trusted Spread Identity Servers associated with the gateways of the source and destination organizations. More specifically, a *Source Spread Identity Server (SSI)* determines the associations of host internal (private) and external (public, routable) IP addresses within the source organization. The SSI includes the functionality of a DNS resolver for the source organization. Traffic flowing in or out of the source organization is processed by a *Source Spread Identity Gateway (SIGS)*, which includes a firewall and router whose NAT entries come from the SSI.

Similarly, a *Destination Spread Identity Server (DSI)* determines the associations of host names and routable IP addresses within the destination organization. The DSI includes the functionality of the authoritative DNS server for the destination organization. Traffic flowing in or out of the destination organization is processed by a *Destination Spread Identity Gateway (SIGD)*, which includes a firewall and router whose NAT entries come from the DSI.

The SSI is a single trusted logical component which includes a modified DNS resolver, and the DSI is a single trusted logical component which includes a modified

authoritative DNS sever. In designs with multiple gateways per organization, the SSI and DSI coordinate the gateway routers and firewalls.

3.2 Assumptions

To ensure appropriate performance and reliability, we assume the following. (1) Each organizational gateway can perform network table translation at link speed [8, 9]. This capability enables each gateway to maintain per-flow state and to translate addresses for each incoming and outgoing packet. (2) Gateways are replicated for high availability. (3) To avoid single points of failure, organizations employ multi-homing [6], with multiple addresses for network gateways and multiple links.

3.3 Protocols

Figure 3 shows how a source S establishes a connection with a destination D, and transfers data, using spread identity. Upon receiving a DNS query from the source, SSI dynamically assigns an identity (external IP address) S*e* to S and forwards a DNS request to DSI. The DSI dynamically assigns an identity D*e* to D's hostname for the DNS response and sends a corresponding NAT entry to SIGD. Based on the DNS response from DSI, SSI sends a NAT entry to SIGS and forwards the DNS response to S. Once this connection is thus established, S sends data packets to D via SIGS and SIGD. An eavesdropper listening to packets sent between SIGS and SIGD sees only the dynamically assigned IP addresses S*e* and D*e*.

We now explain each step of the Connection Establishment and Data Transfer Protocols in more detail.

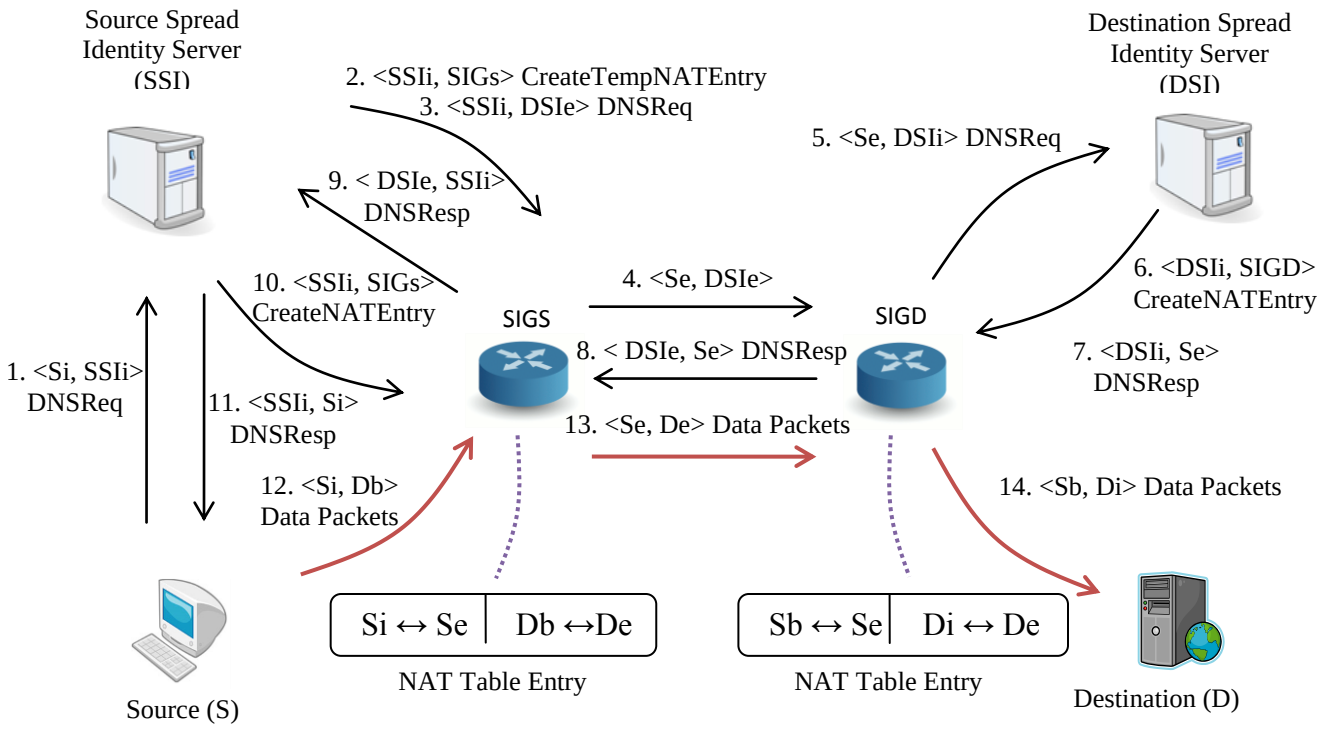

Figure 3 Connection establishment and data transfer protocol for spread identity. When a *Source (S)* establishes a connection with a *Destination (D)*, the *Source Spread Identity Server (SSI)* dynamically assigns an external IP address S*e* to S, and the *Destination Spread Identity Server (DSI)* dynamically assigns an external IP address D*e* to D. The DSI also creates a NAT entry for D*e* at the *Destination Spread Identity Gateway (SIGD).* After receiving a DNS response from DSI, SSI creates an NAT entry for S*e* at the *Source Spread Identity Gateway (SIGS)* and forwards the DNS response to the source. This figure shows the eleven steps (shown in black) that precede the main data transfer (shown in red). For each step, the notation <*s*, *d*> denotes the original source and ultimate destination IP address of the message. For example, in Step 12, the source sends data packets from its internal IP address S*i* to the destination's external IP address D*e* via the SIGS and SIGD. An network eavesdropper listening to packets between SIG_S and SIG_D sees only the dynamically assigned IP addresses S*e* and D*e*.

Throughout, we use the following convention: "[S, *s*] → [D, *d*]: msg" means message msg is sent from S (where *s* is the original source IP address of msg) to D (where *d* is the ultimate IP address for msg). Figure 3 abbreviates this notation as "<*s*, *d*> msg".

We also use the following notation to describe the internal (private) and external (public) IP addresses of various entities. For any entity E, let E*i* denote the internal IP address of E, and let E*e* denote the external IP address of E. Similarly, E*b* is an internal blinding address for E. Thus, SSI*i* denotes the internal IP address of SSI, and DSI*e* denotes the external IP address of DSI.

Connection Establishment Protocol

1. [S, S*i*] → [SSI, SSI*i*]: DNS request (to resolve the hostname of D)

   To communicate with the destination host, the source host needs to resolves the hostname of the destination to obtain the dynamic identity (external IP address) D*e* of the destination.

2. [SSI, SSI*i*] → [SIGS, SIGS*i*]: Request to create temporary NAT entry

   First, the SSI randomly selects a public IP address S*e* for S from the pool of all public IP addresses for its organization that are not currently communicating with the same destination D. Second, the SIGS creates the temporary NAT entry (SSIi ↔ S*e*, DSI*e* ↔ DSI*e*) to be used for the DNS request and response.

3. [SSI, SSI*i*] → [SIGS, DSI*e*]: DNS request

   First, the SSI obtains the IP address DSI*e* of the destination domain authoritative DNS server using its local DNS cache or by making an iterative DNS query to

higher level DNS servers. Second, the SSI sends the DNS request to DSI via SIGS.

4. [SIGS, S*e*] → [SIGD, DSI*e*]: Forwarded DNS request

   Using the temporary NAT entry, the SIGS replaces the source IP address SSI*i* with S*e* in the DNS request. Then, acting as an edge router, the SIGS forwards the modified DNS request to the DSI.

5. [SIGD, S*e*] → [DSI, DSI*i*]: Forwarded DNS request

   The SIGD forwards the DNS request to DSI.

6. [DSI, DSI*i*] → [SIGD, SIGD*i*]: Request to create NAT entry

   First, the DSI randomly selects a public IP address D*e* for D from the pool of all public IP addresses for its organization that are not currently communicating with the same source S. Second, the DSI selects an internal "blinding" address S*b* for S to hide the relationship between S*e* and D*i* within D's private network (for more details, see Section 3.4). Third, the SIGD creates the NAT entry (D*i* ↔ D*e*, S*b* ↔ S*e*) to be used for communications between S and D.

7. [DSI, DSI*i*] → [SIGD, SIGD*i*]: DNS response

   The DSI sends its DNS response (of the resolved IP address D*e*) to S via SIGD.

8. [SIGD, DSI*e*] → [SIGS, S*e*]: Forwarded DNS response

   Using the NAT entries, the SIGD replaces D*i* with D*e* in the DNS response and forwards the modified DNS response to S*e*.

9. [SIGS, DSI*e*] → [SSI, SSI*i*]: Forwarded DNS response

   Using the temporary NAT entry, the SIGS replaces S*e* with SSI*i* in the DNS response and forwards it to SSI.

10. [SSI, SSI*i*] → [SIGS, SIGS*i*]: Create NAT entry request

    First, the SSI selects a blinding address D*b* for D to hide the relationship between S*i* and D*e* within S's private network. Second, the SSI sends a request to SIGS to replace the temporary NAT entry with (S*i* ↔ S*e*, D*b* ↔D*e*) to enable communications between S and D.

11. [SSI, SSI*i*] → [S, S*i*]: Forwarded DNS response

    The SSI forwards the DNS response (of the resolved IP address D*e*) to S.

Now that the connection has been established, S and D can communicate with each other as follows.

Data Transfer Protocol

12. [S, S*i*] → [SIGS, D*b*]: Data packets

    Source S sends data packets to D addressed to D*b*.

13. [SIGS, S*e*] → [SIGD, D*e*]: Data packets

    Using its NAT entries, the SIGS replaces S*i* with S*e* and D*b* with D*e*. Then, SIGS forwards the data packets to D*e*.

14. [SIGD, S*b*] → [D, D*i*]: Data packets

    Using its NAT entries, the SIGD replaces D*e* with D*i* and S*e* with S*b*. Then, SIGD forwards the data packets to D.

3.4 Design and Performance Issues

In this section we briefly identify some important design and performance issues and outline strategies for dealing with them. These issues include scalability, DNS caching, DNS traffic, NATing, communication delays, and address blinding.

Scalability

Within an organization, to achieve scalability and to avoid bottlenecks and single points of failure, it is important to deploy multiple gateways and SSI servers. Following a design proposed by Bellovin *et al.* [10] for distributed firewalls, the SSI (similarly, DSI) can serve as the central policy maker who creates dynamic NAT entries, which are pushed to the SIGS's on the network edges. We assume standard security mechanisms are implemented for securing communications between the SSI and SIGS's.

The single logical unit of the SSI (similarly, DSI) can be implemented with multiple machines with separate internal IP addresses. For example, the SIGD's can balance loads by forwarding DNS traffic to the multiple DSI's in a round-robin fashion.

DNS Caching

Spread identity imposes special challenges to DNS caching stemming mainly from the fact that dynamic address translations are valid only for the duration of the connection. In our architecture, each spread identity gateway maintains each NAT table entry until either its *Time To Live (TTL)* field from the DNS response expires or the associated connection terminates. Whether or not DNS caching makes sense depends on the TTL value and its relationship to the connection duration.

In variations of spread identity in which dynamic translations hold for longer time periods that the TTL value, some forms of DNS caching are possible. In this context,

caching DNS responses at an intermediate hierarchical DNS server works well. Caching by a user application, however, is complicated by address blinding. To deal with this complication, each spread identity gateway could also cache a per source mapping of destination hostname to its blinded address. Tradeoffs are possible between the duration of the dynamic translations and performance related to the amount of DNS traffic.

NATing, DNS Traffic, and Communication Delays

We adopt standard mechanisms for maintaining NAT entries, such as those described by Oskar [11]. Gateways maintain each NAT entry until the TTL field of the DNS response expires or until the connection terminates or times out. We suggest adopting the same time thresholds for our source NATing as is current in effect for traditional Internet gateway NATing.

Criteria for maintaining NAT table entries at spread identity gateways can affect DNS traffic: Higher TTL values in the DNS response decrease DNS traffic but increase the number of NAT entries. We have not measured connection establishment delay.

Spread identity has little impact on communication delay. The main danger for possible communication delay is NATing at the spread identity gateways, but with high-speed firewalls (*e.g.*, by Cisco [9]), NATing can be performed at link speeds.

Address Blinding

The sole purpose of address blinding is to obscure the relationship between source and destination hosts against an eavesdropper who can, for example, read network

traffic within the source organization and who also knows the dynamic remapping of the destination host. Section 5.2 explores this situation further.

Chapter 4

RELATED WORK

We review previous related work in the areas of *Distributed Denial of Service (DDoS)* defense, anonymity, and next generation Internet architectures.

4.1 DDoS Defense

There are three main approaches for DDoS defense: overlay, filter, and capability. Designed using core infrastructure routers, filter-based and capability-based approaches require changes in core infrastructure routers and client software. By contrast, by delegating functionality to a richly-deployed large-scale overlay network, overlay approaches do not require any changes in the core infrastructure, but they can cause significant communication latency [16, 23].

4.1.1 Overlay Approaches

Secure overlay service [12] is the first proposed overlay-based DoS defense mechanism. Only authenticated source traffic is forwarded to the destination host through a series of overlay nodes. Despite the facts that overlay nodes are richly deployed and the destination host is accessible only through the overlay nodes, a DoS attack can be mounted by spoofing the identities of overlay nodes. WebSoS [13] is an overlay approach for web servers, wherein sources are not authenticated. Instead, graphical Turing tests attempt to differentiate attack bots from humans. Stavrou, *et al.* [13] and Wang, *et al.* [23] show that overlay networks increase overall communication network latency by a factor of 5 to 10, due to the underlying chord

routing protocol [53]. Mayday[16] discusses various overlay design choices, allowing a performance-security tradeoff. Stavrou, *et al*. [17] propose a sweeping DoS attack against overlay networks, wherein the attacker follows legitimate source traffic and brings down all overlay nodes with which the source communicates. To defend against such attacks, the multipath overlay approach [17] randomly spreads traffic across multiple overlay nodes.

Stoica, *et al.* [18] proposes a DoS-resilient architecture using the Internet Indirection Infrastructure (I3) based overlay network [19], in which the destination host can dynamically install triggers at overlay nodes to enable communication with legitimate sources. These dynamic triggers are similar to the dynamic NAT entries at the destination in spread identity. Unlike spread identity, however, I3 based approach uses a static destination identity, enabling an attacker to bypass the overlay network once the attacker discovers this static address.

To resist large DDoS attacks, overlay approaches require a rich deployment of the overlay network. In addition, overlay nodes are not managed by a single authority, complicating security management and increasing the risk that the attacker can compromise at least one overlay node. OverDoSe [21] addressed this issue by separating the source and destination hosts at the IP level. In OverDoSe, the destination host uses RSVP-TE [22] to establish tunnels with overlay nodes, which enable destination hosts to teardown connections with compromised overlay nodes dynamically. In addition, OverDoSe uses hash-based computation puzzles to enforce fairness in the request channel.

Akamai's SiteShield [20] is a commercial overlay-based DDoS protection mechanism. In SiteShield, the destination host is accessible only through Akamai's overlay nodes. Because of their large numbers and powerful hardware, overlay nodes in SiteShield can absorb large DDoS attacks. To reduce communication latency, SiteShield uses proprietary overlay routing protocols and web caching. Nevertheless, Akamai's approach requires a large DNS infrastructure.

Unlike overlay approaches, spread identity does not incur high communication latency.

4.1.2 Filter-Based Approaches

When attack traffic surpasses a specified threshold, filter-based approaches install source IP filters. Pushback [25] identifies attack traffic flow and recursively installs filters near the source. Pushback, however, suffers from strategic filter-request spoofing attacks, whereby the attacker attempts to cause legitimate source traffic to be blocked. AITF [54] proposes a three-way handshake protocol to address filter-request spoofing. But AITF suffers from filter-exhaustion attacks, in which the attacker floods the source gateway with filter requests so that the source gateway cannot accept legitimate filter requests. StopIt [55] resists such filter-exhaustion attacks by verifying filter requests using a flow cache before filters are installed. Furthermore, StopIt uses two-level hierarchical fair queuing as a failsafe method to mitigate DDoS attacks against its control channel for sending filter requests.

Filter-based approaches require changes to core infrastructure routers, and they require substantial state information since attacks can come from any source. By contrast, spread identity does not require any changes to routers, and by virtue of

destination filtering requires less state information. Moreover, spread identity destination filtering is less likely to block legitimate traffic.

4.1.3 Capability-Based Approaches

Anderson, *et al.* [26] proposed the first capability-based approach for mitigating DoS attacks. In this two-step approach, the source first sends a request to the receiver seeking permission to send data. Second, if the receiver verifies the sender as a legitimate communicant, the receiver provides an authorization token. The sender includes this token in subsequent data packets, and routers verify the token. This design, however, does not stop DoS flooding attacks of capability requests. SiFF [27] prevents capability flooding attacks on bottleneck links by differentiating legitimate traffic from capability-request traffic. Yet in SiFF, the attacker can still flood the capabilities-request channel. TVA [28] addresses this problem using hierarchical fair queuing based on source path identifieries, but according to Portcullis [29], path-identifier fair queuing scales poorly for large networks. Consequently, TVA proposes per-computation fairness using hashing puzzles.

Filter and capability defenses to DDoS attacks require changes to core infrastructure routers and end hosts. To perform at their best, they also require cooperation among different ISPs. These are among the reasons why, currently, Internet infrastructure providers commonly deploy overlay-based CDN approaches such as Akamai and simply enhance network bandwidth and server resources.

4.2 Anonymity

A variety of approaches have been proposed for achieving anonymity in network applications. Chaum's [7] Mixnet relays each message through a network of one or more mix nodes. At each stage, the current mix node encrypts the message using the next mix node's public key. Each mix node decrypts the received message, removes header information, appropriately pads the message, possibly batches messages, and forwards the message to the next stage.

Other relay-based anonymity techniques fall into two categories: those that introduce large and variable latency (*e.g.,* Babel and Mixminion), and those for interactive applications such as web browsing and SSH that do not introduce significant latency (*e.g.,* Anonymizer, Tor , and Crowds).

Anonymizer [34] removes user identifying information from HTTP requests, changes the source IP address, and forwards the request to the appropriate web server. It is similar to a Mixnet with one mix node, which enables a passive eavesdropper to link sender and receiver. Java Anon Proxy (JAP) [37] solves this problem using a cascade of mixes. PipeNet [38] is a theoretical model for accessing web servers over the Internet. As does JAP, it uses a cascade of mixes. In PipeNet, all clients send legitimate or dummy traffic to the same cascade mix at the same time. Pipenet provides a strong level of anonymity and protects against traffic-analysis attacks, but PipeNet suffers from DDoS vulnerability and inefficiency.

Crowds [36] provides sender anonymity and sender-receiver unlikability by probabilistically relaying web requests to a randomly selected node in the crowd or to the final destination. Replies are sent through the established route. As suggested by

Diaz *et al* [39], the anonymity in Crowds depends on the adversary being unable to observe all links. Hordes extends Crowds and improves its performance by using a UDP proxy and by using multicast replies instead of traversing the reverse path.

Tor [35] provides sender and receiver anonymity based on Chaum's Mixnet. Tor does not make significant attempts to prevent global adversary or traffic analysis attacks. Tarzan [40] is similar to Tor for peer-to-peer anonymous IP overlay networks. Tunnel failures are more frequent in Tarzan because of peer failure or a peer leaving the overlay network. Tunnel failures result in significant computational overhead and latency.

Spread identity provides network anonymity in a fashion similar to that of a mixnet with a fixed path of two nodes. The fixed path avoids the need to insert routing information into messages. As is true for ISDN mixnets [41], the fixed path also protects spread identity from whole set of intersection attacks [42]. Unlike Tor, spread identity avoids overhead of performing multiple encryptions and decryptions, but spread identity does reveal the source and destination organizations.

Spread identity is transparent to the application; thus, there is no need to modify client software. Furthermore, applications with different transport protocols can share communication sessions. As discussed in Section 5.2, spread identity anonymity achieves forward secrecy because once the communication session terminates, SIG gateways (the mix nodes) destroy NAT entries. Unlike Tor, because spread identity transparently anonymizes DNS requests, spread identity does not need to take special measures to prevent DNS leaks [43].

4.3 Other Related Work

TRIAD [44] is a next generation Internet architecture based on content routing that provides scalable content routing, caching, virtual private networking, policy-based routing, and load balancing, without relying on DNS servers. Both TRIAD and spread identity use NATing at source and destination edge routers to translate between external and internal IP addresses. Spread identity works at the IP layer, whereas TRIAD works at the content layer. Unlike spread identity, TRIAD does not map identity (URL) to IP address dynamically.

Both VNAT [45] and spread identity use address blinding in network address translation. VNAT, however, uses address blinding to achieve host mobility, whereas spread identity uses it for anonymity.

Amazon's Elastic Cloud (EC2) reportedly uses address pooling in its load balancing service.

Chapter 5

ADVANTAGES OF SPREAD IDENTITY

5.1 DDoS

We will first discuss our assumptions and threat model, before we discuss different attacks and mitigation techniques to prevent them using Spread Identity architecture.

5.1.1 Threat Model and Assumptions

We assume that source and destination spread identity servers (SSI, DSI) and gateways (SIGS, SIGD) are trusted. Attacker cannot compromise them to launch strategic DDoS attacks against the system. We assume that destination host can detect DDoS attack, using standard techniques like [46]. We also assume that destination host can prevent DDoS attack which exploits application specific vulnerability to consume CPU and memory resources of destination host. For example, there are standard methods like TCP SYN cookie [56] to mitigate these types of attacks.

The goals of attacker include consuming network bandwidth of destination host, DNS servers, and spread identity gateways. Attacker also aims to exhaust secondary resources like CPU, memory of the destination host, spread identity servers, and spread identity gateways. We assume that adversary can create and launch synchronized DDoS attack from millions of attack bot, using tools like Trinoo [47]. Destination host and intermediate routers can be compromised by the adversary. Compromised host/routers can eavesdrop, inject, modify, and discard the traffic. Moreover, adversary can spoof any IP address, while launching DDoS attack.

5.1.2 Bandwidth clogging attacks

In bandwidth clogging attack, master bot machine makes DNS query and obtains the IP address of the victim host. Master machine instructs slave bot for sending attack traffic towards victim machine. Under such attack, network bandwidth of victim gets colluded with attack traffic.

In spread identity architecture, in order to communicate with the destination, source requires access token (in the form of NAT entry at SIGD). In bandwidth clogging attack, slave bots will not have access tokens; therefore, all attack traffic will be dropped at the organization gateways. But, with this preventive measure, attack traffic collude bottleneck link which connects organization network to the Internet. Therefore, SIGD either request ISP's upstream router or SIGS to install destination IP address based filters. Filtering based on destination IP address is efficient, because it requires least state maintenance at upstream ISP's router or at source spread identity gateways (SIGS).

Session hijackings along with source IP address spoofing. An eavesdropper first finds out legitimate access token (i.e. valid NAT entry). It then instructs slave bots to spoof the source IP address of access token and launch DDoS attack to collude bottleneck link of the victim's organization. We can use above preventive measure to stop such attack.

5.1.3 Attacks against the Spread Identity Architecture

We will now discuss strategic DDoS attacks targeted against spread identity architecture, which includes consuming memory, CPU, and network resources of spread identity servers and gateways.

Flooding attack against Destination Spread Identity Server and Gateway: Attacker instructs slave bots to send DNS request to destination spread identity server (authoritative DNS server), thereby creating lot of NAT table entries at destination spread identity gateway (SIGD). Once memory resources of SIGD are consumed, legitimate users will not able to communicate with the destination hosts.

Source IP address based fair queuing can be employed to enforce fairness. But, it does not work well because of IP spoofing and source NATing. Therefore, in current state-of-the-art, graphical turing test [48] is used to separate of attack traffic from legitimate user traffic. Destination spread identity server will create NAT table entry, only when requesting source passes the graphical turing test. There are certain applications (like Web crawlers), which have no involvement of human entity. For these types of applications, we can use hash-based computational puzzle to enforce fairness, as suggested in [21] and [29]. In addition, trusted applications can use pre-shared secret to authenticate with destination spread identity server. NAT entry is created at SIGD only when client provides required authentication credentials.

Bandwidth clogging attacks against DNS servers (Destination spread identity server SIGD): With graphical turing test and hash-based puzzle mechanisms in place, attacker can still launch bandwidth clogging attack against DNS server. One strategy to prevent such attack is to assign wide range of IP addresses to authoritative

DNS server and installing destination IP based filters at upstream ISP's router. In this strategy, even though some of the IP addresses assigned to DNS server are blocked, DNS server is still accessible through rest of the IP addresses.

Flooding attack against Source Spread Identity Server and Gateway: Similar to flooding attack against destination spread identity server (DSI), attack can be launched to flood the NAT table entries at source spread identity gateway by sending flurry of DNS request to source spread identity server (DNS resolver). This attack will consume memory resource of SIGS, which prevents legitimate user's communication. Source spread identity server can throw graphical turing test or hash-based puzzle to the source, before forwarding DNS request to authoritative DNS server. This type of flooding attack is easy to deal with, because attack is originating from the same administrative domain. Therefore, network administrator can easily find out attacking machines and patch them.

5.1.4 Flash crowds

Flash crowd is sudden increase in legitimate user traffic to a particular destination, which results in to increase in packet loss and congestion. We consider detection and prevention of flash crowds as different problem. Therefore, we do not provide any specific solution to flash crowds, but one can employ CDN-based [49] approach.

In this Section, we have discussed broad range DDoS attack and their protection mechanisms provided by spread identity architecture. We have also considered various strategic DDoS attacks against spread identity infrastructure and provided mitigation techniques.

5.2 Anonymity

Spread identity architecture provides sender and receiver anonymity, and sender-receiver unlikability. We discuss various attacks against spread identity anonymity and provide detailed analysis of how they affect user's privacy.

5.2.1 Threat model and Assumptions

We assume that spread identity will be implemented at both ends of communication, moreover source and destination spread identity servers and gateways are trusted. As unencrypted DNS queries reveal the mapping of destination IP and hostname, we assume that DNS queries are encrypted using [50].

Attacker goals include linking the messages to the sender or receiver, and finding out pair of communicating end hosts. To achieve these goals, attacker can compromise en-route routers, and multiple source and destination end hosts. Compromised routers and end host, can eavesdrop, inject, modify, and discard legitimate traffic. In additions, attacker can perform source IP address spoofing to launch various strategic attacks.

5.2.2 Overview of anonymity using Spread Identity

For each session between a source host and a destination host, the source gateway dynamically assigns a temporary pseudonym to the source host chosen as one of the routable IP addresses assigned to the source organization. Similarly, the destination DNS server assigns a temporary pseudonym to the destination host chosen as one of the routable IP addresses assigned to the destination organization. Thus, an eavesdropper listening to packets flowing through the Internet between the source and

destination gateways learns only the source and destination domains; the eavesdropper cannot see the actual complete IP addresses of the source and destination hosts. Although the eavesdropper can link source and destination packets within any session, she cannot link packets between different sessions. Encrypting DNS requests and responses at gateways protects the established bindings of pseudonyms from the Internet eavesdropper.

Our system achieves sender anonymity, because source spread identity server performs dynamic network address translation and transform source internal IP address to public source IP addresses, using NAT table entry (Si ↔ Se, Db ↔ De), as discussed in Section 3.3. Therefore, multiple source machines (Si) can be multiplexed onto single public IP address Se. Similarly, receiver anonymity is achieved by dynamically returning different IP addresses in the DNS response from the pool of available IP addresses. In addition, dynamic network address translation performed at destination spread identity gateways (SIGD), using NAT entry (Di ↔ De, Sb ↔ Se), enable multiplexing of destination IP address amongst different destination host, as long as source IP addresses are different.

Anonymity provided by our system is similar to mixnet [7] with two mix nodes. To create more confusion and make traffic analysis attack hard, we extend our design to use multiple cascaded mix nodes (SIG). For example, with two cascaded mix nodes at source side spread identity, SSI can install following NAT entries (Si ↔ X, Db ↔ Y) and (X ↔ Se, Y ↔ De) at $SIGS_1$ and $SIGS_2$, respectively. Similarly, we can install cascaded spread identity gateways at destination side. With our current design, identity of end host is obscured amongst the range of IP addresses assigned to

the organization. But, implementation of spread identity architecture at ISP level will increase this range to all IP addresses allocated to the ISP, thereby increasing the size of anonymity set.

5.2.4 Security analysis of Spread Identity anonymity.

This sub-section discusses various strategic attacks which attempt to break the sender and receiver anonymity, and finding out communicating sender-receiver pair. We have categorized these attacks as passive and active attacks.

Passive traffic analysis attacks.

Although data traffic is encrypted, attacker can follow particular message through the network by matching exact bits of encrypted payload at every hop on the path. This exact-bit linking attack can be made futile by re-encrypting message at each hop on the path (i.e. Source → SIGS → SIGD → Destination). Other variation of bit linking attack is invading sender-receiver link using message length. Padding all the messages to standard size prevents this type of side channel attack. Global eavesdropper measures packet flow rate at each hop to link sender and receiver, because with high probability each sender-receiver pair exchange information at different packet rates. Our spread identity mechanism uses traffic shaping techniques like inserting dummy traffic to make such traffic analysis difficult. However, as discussed in [52], introducing dummy traffic is not a good solution because it causes network bandwidth inefficiency and degrades overall performance. Finding good strategy to counter traffic analysis attack with good performance is an open research problem.

Active attacks.

Active adversary can specifically mark the message by modifying the few bits to enable traffic analysis, thereby linking sender-receiver pair. Application level integrity checking using hash or messages authentication codes can avoid the tagging attack. Malicious destination host can send flurry of reply messages to perform traffic analysis, thereby invading sender anonymity. In addition, malicious source can reduce the effort of linking sender-receiver flow, by linking a) its own internal IP address with external IP address (Si → Se) b) destination blindfolding address with external destination IP address (Db → De). Active adversary could capture and replay legitimate traffic between particular source-destination pair (Si ↔ Db), to link sender-receiver (<Si, Db> ↔ <Se, De> ↔ <Sb, Di>). These forms of active traffic analysis attacks can be prevented using traffic shaping techniques, like sending dummy traffic.

5.3 Spread Identity implemented only at destination side (Fail-Safe mechanisms)

Earlier we made assumption that source and destination spread identity will be implemented. In this section, we will relax that assumption and discusses benefits of destination only spread identity.

5.3.1 Anonymity

Destination only spread identity will not achieve sender anonymity, because source organization is using static mapping of IP addresses. But, receiver anonymity is achieved, because destination host will be using multiple public IP address (multiple identities) for communication with outside world.

5.3.2 DDoS prevention

Using destination only spread identity, destination hosts will be using multiple identities for communicating with outside world. Therefore, under bandwidth colluding attack, destination can install destination IP address based filters at ISP's upstream router to block attack traffic. Our experimental simulation demonstrates that destination only spread identity mechanism protects bottleneck link between ISP and the organization, under DDoS attack.

Hence, spread identity architecture provides incentive for early deployment, because organization employing spread identity benefits from receiver anonymity, and DDoS attack protection.

Chapter 6

EXPERIMENTAL EVALUATION

6.1 Purpose

We performed experimental evaluation of DDoS defense capabilities of spread identity and compared it with other DDoS defense mechanisms, using ns-2 [4]. We considered capabilities and filter based DDoS protection approaches which include StopIt [55], AITF[54], Pushback[25], TVA[28], and Portcullis [29]. We have not compared our approach with overlay based approaches (WebSoS, Akamai, etc) because its effectiveness increases with increase in number of overlay nodes. Hence, we cannot fairly compare our approach with overlay based approaches.

6.2 Method

Our experimental evaluation measures file transfer success ratio and file transfer time for legitimate users under bandwidth flooding DDoS attack, which attempts to flood bottleneck network link which connects organization to the Internet. We have not performed simulation of NAT exhaustion attack, because it can easily be mitigated by rate limiting DNS request using CAPTCHA or hash based puzzles.

Topology: For realistic simulation, we took similar approach as StopIt [55] to create network topology using BGP table dumps which are obtained from routeview servers [57]. BGP table dumps contain 26K Autonomous Systems (AS) and AS level paths

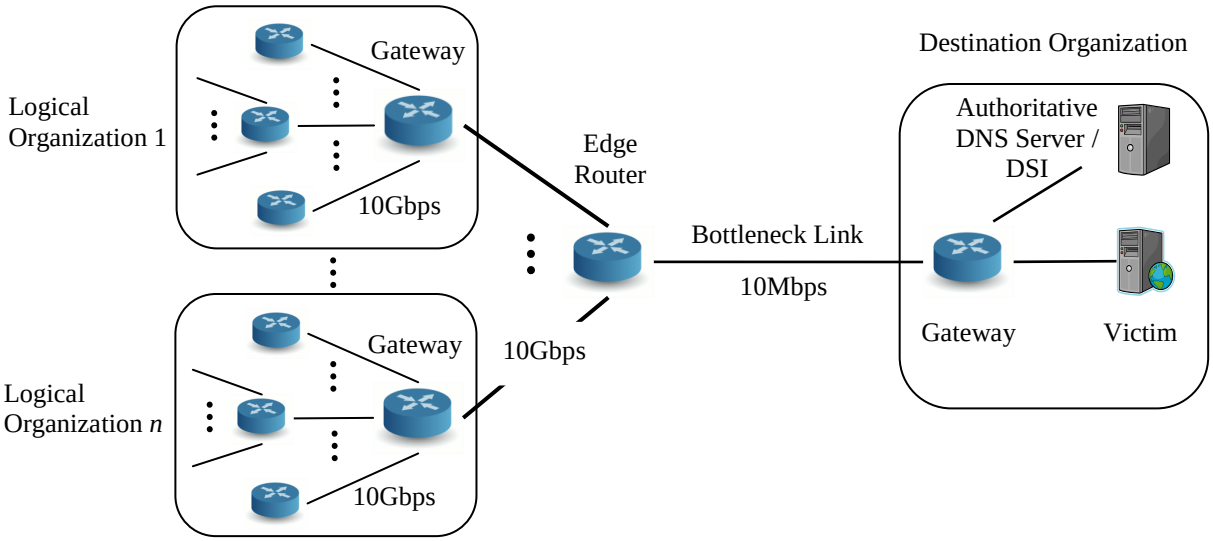

Figure 4 Experimental topology. This figure illustrates network topology used for ns-2 simulation. Destination organization contains authoritative DNS server and Victim, and it is connected to the Internet using bottleneck link of 10Mbps. All routers, which share same gateway for connecting to destination organization, are aggregated into one logical organization. Routers directly connected to the edge router acts as organization gateway of corresponding logical organization.

in the Internet. Our simulator (NS-2) can approximately simulate up to 2000 nodes, hence we randomly sample AS level paths from BGP dumps to create AS level topology. Figure-4 shows our simulated network topology, wherein destination organization is connected to the Internet using bottleneck link of 10 Mbps. In the experiments, we vary the bottleneck bandwidth from 10Mbps to 1Gbps. In order to create organization level topology, we assume all routers, which are connected to edge router, acts as source organization gateways. We aggregate routers in to one logical organization, which contains same gateway in their path to the destination organization.

In our simulation, we vary number of attackers from 1000 to 10 millions. We assume that 1 million attackers can completely collude 1Gbps of bottleneck link. As our simulator (ns-2) supports limited number of nodes, we simulate millions of attackers by changing attack packets interval time with respect to number of attackers. We measure attack packet interval as follows.

$$\text{Packet interval} = \frac{\text{Attack packet size}}{\left\{ \frac{\text{Number attackers} * \text{bandwidth per attacker}}{\text{Number attacker node in the simulator}} \right\}}$$

For example, 1 millions attackers with 10Kbps network bandwidth per attacker have ability to completely collude 10Gbps of bottleneck link. In our simulation, number of attackers were 60 (out of 2000), hence we setup attack packet interval rate to 4.8 microseconds to simulate 1 million attackers with 100 bytes of packet size.

Metrics: To measure effectiveness of spread identity DDoS protection, we used two metric a) success ratio of file transfer b) file transfer time.

$$\text{Success Ratio} = \frac{\text{\# of successful file transfers}}{\text{\# of successful file transfers + \# file aborts}}$$

Similar to StopIt [55], in our simulation, legitimate users start TCP file transfer of 20KB file. To finish our simulation in reasonable time, we setup file transfer time out to be 25sec.

Implementation: We have implemented spread identity architecture and DNS protocol in ns-2. In our experiments, we used two versions of spread identity, a) SI+:

implements mutually trusted source and destination spread identity gateways b) SI: implements destination only spread identity. With SI+, destination spread identity gateway (SIGD) can install filters at source spread identity gateways (SIGS). On the other hand, SI installs filters at ISP's upstream routers.

Pushback [25] DDoS protection mechanism is officially part of ns-2. We adopted the implementation of other DDoS protection mechanism from StopIt [55], which includes StopIt, TVA, TVA+, AITF, and Portcullis. TVA+ is extension to TVA; it uses Passport [31] authentication mechanism to avoid source IP address spoofing.

6.3 Results and Analysis

Figure 5(a) and 5(b) show observed success ratios and file transfer times for various DDoS protection mechanisms. In the first experiment, we set the bottleneck link at 1Gbps, and legitimate users transferred 20KB files. We varied the number of attackers from 1K to 10M.

The success ratio of portcullis dropped suddenly after 100K attackers with file transfer timeout at 25sec. In portcullis, however, legitimate users can complete their file transfers by waiting longer and solving difficult puzzles. As discussed in StopIt [55], AITF does not perform well for 1 million attackers because it uses a three-way handshake protocol for installing filters. Therefore, under a large volume of attack traffic, SYN-ACK packets are lost. Success ratios of the TVA and pushback DDoS protection mechanisms are similar because they perform per-path-fairness. TVA performs hierarchical fair queuing based on path identifier and Pushback recursively installs filters near the source. Due to per-path-fairness, legitimate users suffer

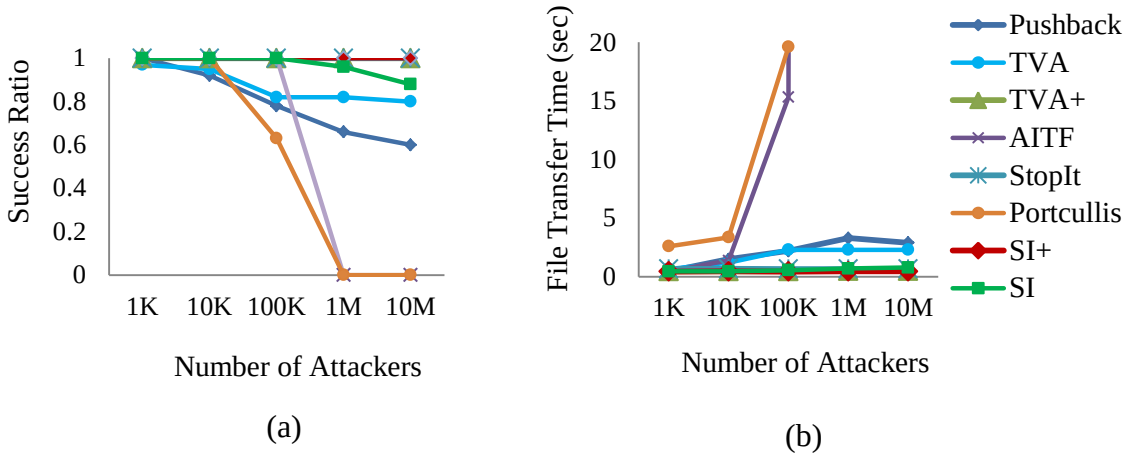

Figure 5 Experimental evaluation for bandwidth-colluding attacks. We used a 1Gbps simulated bottleneck link connecting organizations to the Internet. Legitimate clients executed 20Kb of file transfers with a 25sec timeout. We varied the number of attackers from 1K to 10M and measured: (a) success ratios and (b) average file transfer times, for various DDoS protection mechanisms. SI+ installs destination IP based filters at organization gateways, whereas SI installs filters at edge routers. Overall, our spread identity mechanism performed similarly to StopIT and TVA+, and it outperformed these and all other defenses tested for 10 millions attackers.

because they share the same path with the attackers. StopIt, TVA+, and our Spread Identity mechanism outperform the other DDoS protection solutions.

Figure 6(b) compares the file transfer times for various file sizes. TVA+ was most efficient as compared to StopIt because it does hierarchical fair queuing, thereby giving more preference to legitimate traffic than attack traffic. With our spread identity architecture, destination IP address based filtering stopped attack traffic immediately; hence, only legitimate traffic flowed through bottleneck link.

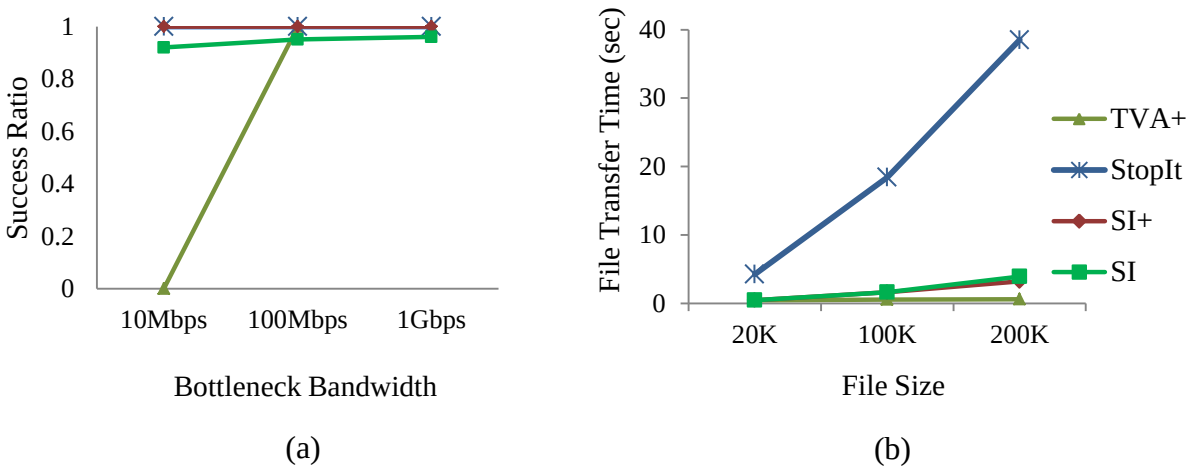

Figure 6 Experimental evaluation of DDoS defense mechanisms for various traffic patterns and bottleneck links. In Figure 6(a) we used 1 million attackers with a 1Gbps simulated bottleneck. Figure 6(b) shows file transfer times for various file sizes. In Figure 6a, we measured success ratios, varying the bandwidth of bottleneck links from 10Mbps to 100Mbps with 1 million attackers. These experimental results show that spread identity is as effective as other filter and capability DDoS protection mechanisms

In another set of experiments, we measured the effectiveness of spread identity as we varied the bandwidth of bottleneck links connecting organizations to the Internet. Figure 6(a) shows StopIt performed well because attack traffic was completely blocked near the source.

The success ratio of TVA+ was 0 for 10Mbps of simulated bandwidth because attackers can flood the bottleneck links with capability request packets. Similar to StopIt, SI and SI+ outperforms.

Chapter 7

DISCUSSION

7.1 DNS caching timeout and size of NAT table at destination spread identity gateways.

In spread identity architecture, destination spread identity gateway maintains NAT entries, until Time to Live (TTL) field of the DNS response is expired or there are no ongoing connections through the NAT entry for certain time period. Our mechanism for maintaining NAT entries creates tradeoff between size of NAT table and TTL field of DNS response. Setting the small value in the TTL field causes more DNS traffic from the clients, because clients will cache DNS response for small amount of period. But, on the other hand, setting TTL field to higher value can cause increase size of NAT table at spread identity gateways.

We have not completely explored the best strategy for this tradeoff. One good strategy could be setting higher value in TTL field when size of NAT table is small, and on the other hand when size of NAT table is higher, spread identity server should set small value in the TTL field. Moreover, organization can deploy a high end server, which has fast computing capabilities and large amount memory, to improve the performance and lessen DNS traffic.

7.2 Other type of strategic DDoS attack against Spread Identity architecture for the Internet.

Source IP address based filtering is infeasible for DDoS attacks with millions of attackers. Using spread Identity mechanism, organization can filter attack traffic based on destination IP addresses, thereby reducing number of filters at upstream routers. But, attacker can launch strategic attack against spread identity architecture by sending attack traffic to all IP addresses of the organization, in round robin fashion. Spread Identity architecture cannot install filters on the entire destination IP addresses, because it will disconnect the organization from the Internet. We are currently exploring prevention technique for such DDoS attack, but impact of such strategic DDoS attack can be lessened by implementing spread identity at ISP level, because ISPs are connected to the Internet using high speed links (OC-192 – 9.6Gbps).

7.3 Open Problems

Directions for further work include the following. (1) Perform additional experiments to measure connection establishment time, communication delays, and DNS traffic. (2) Experimentally compare spread identity with overlay methods. (3) Carry out rigorous security analysis of the protocol strength and of its resulting anonymity properties. (4) Explore additional applications of the spread identity concept, including applying it at the ISP level.

Chapter 8

CONCLUSION

We have presented and experimentally evaluated a new *spread identity* architecture to provide network anonymity and enhanced DDoS defense based on destination filtering with low impact on legitimate traffic. It also provides a limited traceback capability. This architecture leverages trusted organizational gateways and applies fundamental concepts of dynamic bindings, address pooling, indirection, and pseudonyms. The main deployment costs are modifying organizational gateways and increasing DNS traffic, but no changes are required to Internet routers. Furthermore, and unlike overlay approaches, spread identity does not slow down communication traffic appreciably. Our simulations demonstrate that the approach is viable and that our destination filtering works as well as existing approaches based on filtering- and capability-based DDoS mechanisms. Specifically, using the ns-2 simulator, we demonstrate that file transfer success ratios for our spread identity DDoS protection mechanism are similar to those of filter- and capability-based approaches, with lower file transfer times than for filter-based approaches.

Although we focus on spread identity at organizational gateways, the concept is applicable much more broadly. For example, spread identity could be applied at ISP gateways and in many network applications.

REFERENCES

[1] RSA Security Inc, The power behind RSA SecurID: Two-Factor user authentication: RSA ACE/Server, http://www.opsec.com/solutions/partners/downloads/rsa_securid_whitepaper.pdf, Last accessed Nov. 14, 2009.

[2] Molloy, I., Li, J., Li, N.: Dynamic virtual credit card numbers. In: Dietrich, S., Dhamija, R. (eds.) FC 2007. LNCS, vol. 4886, pp. 208–223. Springer, Heidelberg (2007)

[3] Phatak, D. S. 2005. Spread-Identity mechanisms for DOS resilience and Security. In Proceedings of the First international Conference on Security and Privacy For Emerging Areas in Communications Networks (September 05-09, 2005). SECURECOMM. IEEE Computer Society, Washington, DC, 23-34.

[4] S. McCanne, S. Floyd: "The Network Simulator NS-2", URL http://www.isi.edu/nsnam/ns/

[5] Charlie Kaufman, Radia Perlman, and Mike Speciner. Network Security: Private Communication in a Public World. Prentice-Hall, New Jersey, 1995.

[6] H. Wang, H. Xie, L. Qiu, A. Silberschatz, and Y. R. Yang. Optimal ISP subscription for Internet multihoming: Algorithm design and implication analysis. In Proceedings of IEEE INFOCOM ’05, Miami, FL, Apr. 2005.

[7] D. Chaum, Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms, CACM 4 (2), 1982.

[8] Chen, Yuqun, Angelos Bilas,Stefanos N. Damianakis, Czarek Dubnicki, and Kai Li, UTLB: A Mechanism for Translations on Network Interface, ASPLOS8, Oct, 1998, 193-204.

[9] Mario Mazzola, Tom Edsall, Luca Cafiero, Address translation mechanism for a high-performance network switch, U.S. Patent 5740171, Apr. 14, 1998

[10] S. Ioannidis, A. Keromytis, S. Bellovin, and J. Smith. Implementing a Distributed Firewall. In Proceedings of Computer and Communications Security (CCS), pages 190–199, November 2000

[11] Oskar Andreasson : IPTables Tutorial,
URL http://www.faqs.org/docs/iptables/tcpconnections.html

[12] A. Keromytis, V. misra, and D. Rubenstein, SOS: Secure overlay services, in Proceedings of SIGCOMM, Pittsburgh, PA, Aug 2002, pp. 61–72.

[13] Stavrou, A., et al., WebSOS: An Overlay-based System For Protecting Web Servers From Denial of Service Attacks. Elsevier Journal of Computer Networks, special issue on Web and Network Security, 2005.

[14] J. Arkko and H. Haverinen, Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA), RFC 4187, January 2006

[15] A. Stavrou, A. D. Keromytis, J. Nieh, V. Misra, and D. Rubenstein. MOVE: An End-to-End Solution To Network Denial of Service. In Proceedings of the ISOC Symposium on Network and Distributed System Security (SNDSS), pages 81–96, February 2005.

[16] D. Andersen. Mayday: Distributed Filtering for Internet Services. In 3rd Usenix USITS, 2003.

[17] A. Stavrou and A. Keromytis. Countering DoS attacks with stateless multipath overlays. In ACM CCS, 2005.

[18] K. Lakshminarayanan, D. Adkins, A. Perrig, and I. Stoica.Taming IP Packet Flooding Attacks. In Proc. HotNets-II, 2003.

[19] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, Internet indirection infrastructure, in Proceedings of SIGCOMM, Pittsburgh, PA, Aug 2002, pp. 73–86.

[20] Akamai SiteShield Module
URL http://www.akamai.com/dl/feature_sheets/Akamai_Site_Shield.pdf

[21] E. Shi, I. Stoica, D. Andersen, and A. Perrig. OverDoSe: A Generic DDoS Protection Service Using an Overlay Network. Technical Report CMU-CS-06-114, Carnegie Mellon University, 2006.

[22] D. O. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. RSVP-TE: Extensions to RSVP for LSP Tunnels. Internet Draft draft-ietf-mpls-rsvp-lsp-tunnel-08.txt, IETF, February 2001.

[23] J. Wang, X. Liu, and A. A. Chien. Empirical Study of Tolerating Denial-of-Service Attacks with a Proxy Network. In Proceedings of the 14th USENIX Security Symposium, pages 51–64, August 2005.

[24] XiaoFeng Wang, Michael K. Reiter, Using Web-Referral Architectures to Mitigate Denial-of-Service Threats, IEEE Transactions on Dependable and Secure Computing, pp. 203-216, April-June, 2010.

[25] R. Mahajan, S. Bellovin, S. Floyd, V. Paxson, and S. Shenker. Controlling high bandwidth aggregates in the network. ACM Computer Communications Review, 32(3), July 2002.

[26] Tom Anderson, Timothy Roscoe, and DavidWetherall. Preventing Internet denial-of-service with capabilities. In Proceedings of Hotnets-II, November 2003.

[27] A. Yaar, A. Perrig, and D. Song. SIFF: A Stateless Internet Flow Filter to Mitigate DDoS Flooding Attacks. In IEEE Symposium on Security and Privacy, 2004.

[28] X. Yang, D. Wetherall, and T. Anderson. TVA: A DoS-limiting Network Architecture. In IEEE/ACM Transactions on Networking (to appear), 2009.

[29] B. Parno, D. Wendlandt, E. Shi, A. Perrig, B. Maggs, and Y.-C. Hu. Portcullis: Protecting Connection Setup from Denial-of Capability Attacks. In ACM SIGCOMM, 2007

[30] J. Xu and W. Lee, Sustaining Availability of Web Services under Distributed Denial of Service Attacks, IEEE Transactions on Computers, vol. 52, no. 2, pp. 195–208, 2003.

[31] X. Liu, X. Yang, D. Wetherall, and A. Li. Passport: Secure and Adoptable Source Authentication. In Proc. 5th USENIX NSDI, Apr.2008.

[32] Ceki Gulcu and Gene Tsudik. Mixing E-mail with Babel. In Network and Distributed Security Symposium - NDSS '96, February 1996. IEEE.

[33] Danezis, G., Dingledine, R. & Mathewson, N. (2003), Mixminion: Design of a Type III Anonymous Remailer Protocol, in IEEE Symposium on Security and Privacy, Berkeley, CA.

[34] Anonymizer

URL http://www.anonymizer.com/

[35] Dingledine, R., Mathewson, N. & Syverson, P. (2004a), Tor: The secondgeneration onion router, in Proceedings of the 13th USENIX Security Symposium.

[36] Reiter, M. & Rubin, A. (1998), `Crowds: Anonymity for web transactions', ACM Transactions on Information and System Security (TISSEC) 1(1), 66-92.

[37] Hannes Federrath. JAP— Anonymity & Privacy.

URL http://anon.inf.tu-dresden.de/index en.html. Accessed 10 January, 2007.

[38] W. Dai. Pipenet 1.1. Usenet post, August 1996.

URL http://www.eskimo.com/˜weidai/pipenet.txt First mentioned in a post to the cypherpunks list, Feb. 1995.

[39] George Danezis and Claudia Diaz. A survey of anonymous communication channels. Technical Report MSRTR-2008-35, Microsoft Research, January 2008.

[40] Freedman, M.J. and R. Morris, Tarzan: A Peer-to-Peer Anonymizing Network Layer, Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS), Washington DC, USA, 2002.

[41] A. Pfitzmann, B. Pfitzmann, and M. Waidner. ISDNmixes: Untraceable communication with very small bandwidth overhead. In GI/ITG Conference on Communication in Distributed Systems, pages 451–463, February 1991.

[42] Berthold, O., Pfitzmann, A. & Standtke, R. (2000), The disadvantages of free MIX routes and how to overcome them, in H. Federrath, ed., Designing Privacy Enhancing Technologies, Vol. 2009 of LNCS, Springer-Verlag, pp. 30-45.

[43] Jeremy Clark, P. C. van Oorschot, and Carlisle Adams. Usability of anonymous web browsing: an examination of Tor interfaces and deployability. In Proceedings of the 3rd Symposium on Usable Privacy and Security (SOUPS '07), pages 41–51, New York, NY, USA, July 2007. ACM.

[44] D. R. Cheriton and M. Gritter. TRIAD: A new next generation internet architecture.
URL http://www-dsg.stanford.edu/papers/triad/triad.html, Mar 2000.

[45] G. Su and J. Nieh, Mobile Communication with Virtual Network Address Translation, Technical Report CUCS-003-02, Department of Computer Science, Columbia University, February 2002.

[46] V. Sekar, N. Dufeld, J. van der Merwe, O. Spatscheck, and H. Zhang, LADS: Large-scale Automated DDoS Detection System," in Proc. USENIX Annual Technical Conference, 2006

[47] Dave Dittrich. The DoS Project's 'trinoo' distributed denial of service attack tool. Technical report, University of Washington, 2000.
URL http://staff.washington.edu/dittrich/misc/trinoo.analysis.txt

[48] L. von Ahn, M. Blum, N. J. Hopper, J. Langford, CAPTCHA: Using Hard AI Problems For Security, in: Proceedings of EUROCRYPT, 2003.

[49] Jung, J., Krishnamurthy, B., and Rabinovich, M. 2002. Flash crowds and denial of service attacks: characterization and implications for CDNs and web sites. In Proceedings of the 11th international Conference on World Wide Web (Honolulu, Hawaii, USA, May 07 - 11, 2002). WWW '02. ACM, New York, NY, 293-304.

[50] G. Ateniese and S. Mangard. A new approach to DNS security (DNSSEC). In ACM CCS, 2001.

[51] Diaz, C., Seys, S., Claessens, J. & Preneel, B. (2002), Towards measuring anonymity, in R. Dingledine & P. Syverson, eds, Privacy Enhancing Technologies workshop (PET 2002), Vol. 2482 of LNCS, Springer-Verlag, San Francisco, CA, USA, pp. 54-68.

[52] A. Back, U. M¨oller, and A. Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. In I. S. Moskowitz, editor, Information Hiding (IH 2001), pages 245–257. Springer-Verlag, LNCS 2137, 2001.

[53] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. Submission to ACM SIGCOMM, 2001.

[54] X. Yang, D. Wetherall, and T. Anderson, A DoS-limiting network architecture, ACM SIGCOMM Computer Communication Review, vol. 35, no. 4, pp. 241–252, 2005.

[55] X. Liu, X. Yang, and Y. Lu. To Filter or to Authorize: Network-Layer DoS Defense against Multimillion-node Botnets. In ACM SIGCOMM, 2008.

[56] Lemon, J. Resisting SYN floods DoS attacks with a SYN cache. In Proceedings of USENIX BSDCon'2002 (Feb. 2002).

[57] Route Views Archive.

URL http://archive.routeviews.org/oix-route-views/

Appendix A: List of Abbreviations and Acronyms

CDN	Content Delivery Network
DDoS	Distributed Denial of Service
DNS	Domain Name Server
DSI	Destination Spread Identity server
DSI*e*	external IP address of the DSI
DSI*i*	internal IP address of the DSI
IP	Internet Protocol
ISP	Internet Service Provider
NAT	Network Address Table
SSI	Server Spread Identity Server
SSI*e*	external IP address of the SSI
SSI*i*	internal IP address of the SSI
TTL	Time To Live