Tor Instead of IP

Vincent Liu, Seungyeop Han, Arvind Krishnamurthy, Thomas Anderson
University of Washington
{vincent, syhan, arvind, tom}@cs.washington.edu

ABSTRACT

As the Internet has become more popular, it has increasingly been a target and medium for monitoring, censorship, content discrimination, and denial of service. Although anonymizing overlays such as Tor [2] provide some help to end users in combating these trends, the overlays themselves have become targets in turn. In this paper, we take a fresh approach: instead of running Tor on top of IP, we propose to run Tor instead of IP. We ask: what might the Internet look like if privacy and censorship resistance had been designed in from scratch? To be practical, any proposal also needs to be robust to failures, achieve reasonable efficiency compared to today's Internet, and be consistent with ISP economic concerns. Although preliminary, we argue that our design achieves these goals.

1. INTRODUCTION

As the Internet has evolved, some of the original design principles have become deprecated, and others have come into prominence. One example of a design principle that was not included is security. Some claim that the Internet should have been designed with security in mind from the beginning. However, exactly what is meant by security is unclear.

Over the years, many attempts have been made to define a security architecture for the Internet. Most have focused on authentication in order to prevent faking of IPs and routes (Passport [7], SBGP [5], SCION [10]), but in this paper, we ask a different question: what if privacy, not authentication, was our overriding goal?

Censorship has become endemic—the constant battle against hackers has led us to designs where ISPs have the ability to snoop every packet. ISPs have begun to work with the content industry to enforce copyright, by disabling Internet access of anyone who is claimed to be troublesome. Even if this is acceptable, we are, in effect, building in the technical ability for governments and corporations to constrain what we can say and who we can say it to. China is a classic example of a government that already exercises this power, but others willing and able to censor can be found around the world in places like France, South Korea, and even the United States.

For a universal data network, we argue that privacy should be a first principle. In the original phone network, society developed strong legal protections against eavesdropping, but in a data network, eavesdropping is trivial and has silently become standard practice. Commercial interests want to control what we communicate, so that they can preferentially charge for it. Worse, the solutions we might adopt in the future to improve security, such as authenticated, traceable communication, makes censorship and differential control all that much easier.

Privacy is something that we cannot easily add on top of the Internet. Encrypting traffic can protect the contents, but this would still disclose with whom we are communicating to every ISP along the path. Laundering packets through an anonymizing overlay, such as Tor, can conceal the source and the destination, but this is easily circumvented, as governments can blacklist Tor nodes or monitor all Tor exit traffic so that traffic analysis can reveal the source. After all, traffic to or from Tor essentially advertises itself as probably worth tracking.

We take an explicitly clean slate approach. It is not our intent to describe a plausible story for how we might fix the Internet to improve privacy—there are immense government and corporate interests in violating our privacy! But rather to ask the much simpler question: if we had known what we know now, how might we have designed the Internet differently? We will return to discuss possible paths to adoption in §7.

We must be careful when introducing privacy to the Internet as a naive implementation could easily open the system up to other security attacks. To be practical, the system needs all of the other properties we want from a network: to be reliable even if some ISPs are miscreants, to be DoS resistant, to be reasonably efficient, to allow for a market in data delivery services, etc. Our main contribution is to show what we need to sacrifice for privacy—some hardware cost, and some inefficiency in route selection. But we argue our design is feasible.

Our basic approach is that nothing on the wire should reveal any information about the source or destination of a packet, except that it came from/went to its immediate neighbor. Our approach borrows ideas about anonymous routing from Tor, trust models from SCION, and DoS resilience from i3 [9]/Phalanx [3]. We also exclude from our consideration the case where all ISPs collude together to violate privacy, and we consider robustness

against timing analysis to be orthogonal and complementary to our work. Our goal is not to make effective wiretapping impossible, just extraordinarily difficult.

2. RELATED WORK

There has been much progress on the topics of anonymity and network neutrality in the literature. The most relevant project is Tor [2], which uses a peer-to-peer overlay to provide anonymization using Onion Routing. However, Tor-like overlays are easily censored due to their use of centralized lists of relays that are visible to censors. Our approach is similar to Tor in that we use onion routing, but at the level of each ISP. Moving Tor's functionality into the network introduces new challenges.

The Address Hiding Protocol [8] proposes to assign a random IP address (still from the same ISP) to hide the original source from the destination. While AHP can improve the privacy of users, it does not protect against violations of privacy by the users' own ISP. Unfortunately, it is often the users' ISP which is most interested in censorship and network neutrality infringements.

Some work has also been done on DoS resistance, most notably by Lakshminarayanan et al. [6] and later extended with Phalanx [3], which both build on the i3 system [9]. In Phalanx, all packets to a given destination are sent through an intermediate mailbox node, which only forwards packets to the destination when they are specifically requested by the destination. ISPs only allow requested packets to be forwarded to destinations. Phalanx assumes a large number of mailboxes so that, while DoS attacks against mailboxes are possible, no attacker is strong enough to take all mailboxes offline. The system prescribes a rotation of traffic through a random sequence of mailboxes so that even if a subset are DoS attacked, only a fraction of any given flow will be lost.

3. THREAT MODEL AND ASSUMPTIONS

The success of the current Internet is largely due to the end-to-end principle—by decoupling applications and hardware advances, users can adopt new applications without requiring help or changes from the network. However, examples such as censorship or traffic discrimination have shown that there are strong incentives for various entities in the middle of the network to meddle with applications. Our goal is to extend and strengthen the end-to-end model so that, by design, the network is unable to prevent new applications and uses.

In our model, users and services wish to be able to communicate in private, without the ability for third parties or the networks (e.g., acting on behalf of government) to monitor, prevent, or limit that communication.

Attackers can be third parties—other users who might want to monitor, limit, or prevent certain uses of the network. They can do this by querying databases exhaustively, or by launching DoS attacks against end users, services, databases, or ISPs.

ISPs can also launch these attacks. They can also monitor or squelch any traffic that traverses them, and so a key point is that traffic should not provide any information about the contents or destination of packets. Although ISPs may wish to block some types of content or eavesdrop on passing traffic, we assume that they are willing to give users connectivity to portions of the Internet that are outside of their domain—in other words, our design does not consider the case that ISPs block all traffic to/from the outside world.

While multiple attackers may collude in cooperation with each other or the government, we assume attackers only control a portion of the physical resources of the network. Finally, protection against traffic analysis and timing attacks is orthogonal to our work.

4. GOALS AND DESIGN PRINCIPLES

The desired properties of our system are as follows:

- *Anonymity*: The source, destination, and service of any packet should be hidden, except that the first hop ISP will know that it is accepting the packet from/delivering it to one of its customers.
- *Censorship-resilience*: Neither the system nor any part of it should be censorable. This includes specific services/destinations, ISPs, and any control-plane mechanisms like name resolution.
- *Net-neutrality*: ISPs should not be able to discriminate against a specific source or destination.
- *DoS-resilience*: Even when DoS attacks take down specific nodes and links, end hosts should be able to communicate with each other. Not only is this a nice property, it is necessitated by the above goals—DoS is a potential tool for censorship (e.g., against Estonia or Wikileaks), and one would expect that anonymity makes DoS attacks easier.
- *Decentralization*: The system should not rely on or trust any single point of weakness. This is also necessitated by the above properties since centralized solutions are easily censorable and DoSable.
- *Minimal Disruption*: The system should have minimal impact on the current view of the Internet. For example, path dilation should be small, policy control should be available, and pricing should be similar to the current economic model.

5. SYSTEM OVERVIEW

We assume a network layout that is very similar to the structure of today's Internet. We augment this layout

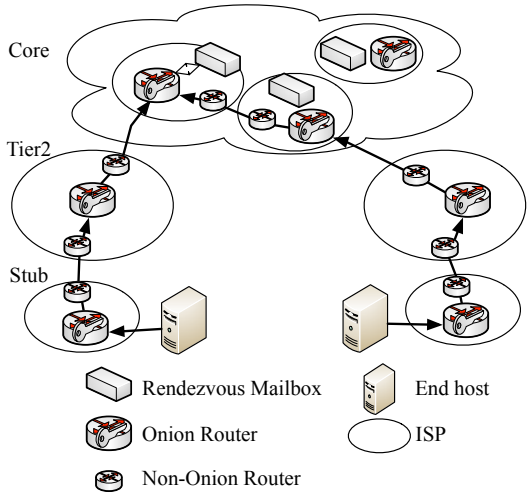

Figure 1: Overview of the Network Layout

with two types of middleboxes: onion routers and rendezvous mailboxes. We also define a designated group of core ISPs through which all traffic is routed. Figure 1 depicts the overview of the network layout in our system, and we will explain each component here.

Onion Routers: Onion routing uses recursive encryption to route through a series of intermediates without them knowing the source or the destination. This has been used in Tor-like systems to provide anonymity, and we require that all traffic travels through these onion routers.

Onion routers provide two functions: circuit extension, which is used to incrementally build an onion circuit and cell relaying, which is the process of forwarding packets through established onion circuits. Both of these functions will be explained in detail in §6.3.

Every ISP contains onion routers that customers and adjacent ISPs utilize for every AS hop, thus creating a world where onion routers are as fundamental to providing connectivity as bandwidth. Note that traffic through an ISP can potentially pass through multiple non-onion routers, but because all elements in an ISP are in the same trust domain, traffic only needs to pass through exactly one onion router per ISP, reducing the performance costs.

The specific onion router used is not important to the end host and not specified in the protocol, so we assume that as soon as a packet enters ISP A, A picks one and internally routes to it. When the end host asks for an extension to a neighboring ISP, A internally routes the packet from the onion router to the appropriate egress.

Rendezvous Mailboxes: An effective method to provide DoS resilience is to prevent senders from sending a significant amount of traffic to a receiver without explicit authorization from the receiver. Rendezvous mailboxes in our system support a put/get interface, where, for a sender to send anything, it needs to send a packet to a rendezvous mailbox rather than directly to the destination. The destination polls the mailbox for the packet, then a brief handshake occurs between the sender and the destination to agree on a sequence of mailboxes over which to communicate. Because hosts can only send (put) packets to mailboxes, and packets are fetched (get) on-demand, the destination cannot be DoSed provided ISPs only deliver packets that were retrieved from mailboxes. The number of rendezvous points prevents an attacker from taking down a given service, and the secret sequence of mailboxes prevents an attacker from significantly disrupting any specific connection.

Each end-host utilizes many mailboxes, where the exact number is determined by the bandwidth and DoS resilience requirements of the end-host, and each mailbox is also able to host multiple end-hosts depending on the capacity of the mailbox and capacity required by each end-host. Mailboxes are placed in core ISPs (explained below) and to obtain them, end hosts route to the core where they query a random ISP for available mailboxes.

Unlike pure onion routing, the layer of indirection provided by mailboxes also increases anonymity for the service host as well as the source in a manner similar to Tor hidden services. Thus, although this deviates from the current structure of the Internet, mailboxes help to provide both anonymity and DoS resilience.

Core ISPs: We define a set of core ISPs that are equipped to handle a large amount of traffic and peer with at least two other core ISPs. Current Tier 1 ISPs would fall into this set, as would some large Tier 2 ISPs.

Since all traffic travels through mailboxes, they should only reside in core ISPs. Additionally, we relax peering polices between core ISPs to allow zigzags (i.e., random, arbitrary, inter-ISP paths among the core) for more anonymity. Onion routing still occurs on the path to the core, but because of routing policies, it is possible that traffic will not be able to follow arbitrary routes until the core, hence requiring multiple hops in the core.

6. PROTOCOL

In the following subsections, we first talk about route discovery in §6.1, and talk about an abstraction called authorities in §6.2. We then explain how paths are set up in our system in §6.3.

6.1 Route Discovery

As we require exactly one onion router per ISP per path, we only need to treat each ISP as a single entity and specify interdomain routing. Because of privacy concerns, an ISP should not be given control over the entire path so we only give them control over the portion of the

path that passes through them—they sign advertisements of the form {ingress ISP, transit ISP (itself), egress ISP} that indicate they are willing to provide transit given an ingress ISP and egress ISP. These small path sections are a modification to pathlets [4] that are constrained to exactly length three. These triplets can be composed with each other provided the second and third ISPs of a pathlet match the first and second ISPs of the next pathlet in the chain. This lets end hosts have control over their paths.

There are two ways to propagate these path sections: through broadcasting, which is used for bootstrapping and general route construction, and through storage in an authority database, which is used when filtering is suspected. Each ISP broadcasts the updates to the rest of the network, and ISPs aggregate this information and share it with their end hosts. The authority database (explained in §6.2) is provided because broadcasted updates are prone to filtering by downstream ISPs.

With such a routing model, IP addresses are not necessary. Instead, end hosts can get to the next hop by specifying either the next ISP's public key, or by specifying a rendezvous mailbox within the current ISP. Rendezvous mailboxes have addresses of the form *P(ISP):identifier*, where *P(ISP)* is the public key of the ISP in which the mailbox resides. Aside from ISPs and mailboxes, nothing else in the system is addressable.

Triples allow for policy-compliant routing and lets smaller ISPs to protect themselves from excess traffic by not advertising transit routes (or any routes not from their end hosts). A typical path will take a path up to the core, where it can optionally zigzag around until it reaches the rendezvous node. Note that a core consisting of Tier 1 and some Tier 2 ISPs implies that source/destination pairs that do not need anonymity will have low path dilation compared to paths in the current Internet since these pairs will take a shortest path to the core and not zigzag. Significant increases in latency and network load only happen for circuits for which anonymity is required.

6.2 Authorities

We assume a small trusted-computing base in the form of a set of well-known root authorities. Authorities can recursively sign other authorities to create a hierarchical chain of trust in a similar fashion as present-day root certificate authorities. They provide a database used for:

- Name resolution for a distinct subset of the namespace so as to not have conflicting information.
- Bootstrapping lists of ISPs and their public keys.
- Disseminate signed ISP routing triplets.

All the goals listed in §4 must also apply to our authority mechanism, and in order to fulfill these properties, different authorities are managed by powers that represent a specific contractual/legal/geographical domain akin to the trust domains of SCION [10]—we do not assume that they are trusted, only that they are not colluding with each other. They also utilize rendezvous mailboxes to protect against DoS attacks and provide a way for end hosts to get unfiltered information from authorities through a secure connection. The addresses of these rendezvous mailboxes are preconfigured within each ISP exactly as DNS server addresses are in the current Internet.

Name Resolution: The name resolution service provides the following mapping: $<name> \rightarrow <P(S),\ P(ISP_R):R>$, where S is the service in question, $P(X)$ is public key of X, and $P(ISP_R) : R$ is the address of one of many rendezvous mailboxes for that service.

These records are signed by authorities in a way that is similar to DNSSEC [1], with a slightly different mapping and without backward compatibility and caching at the level of ISPs. There is a large body of research on properties, attacks, and fixes for DNSSEC; however, these discussions are beyond the scope of our current work.

When a client wants to avoid its ISP knowing its queries, the client needs to connect to an onion router outside of the ISP in the manner described in §6.3 and do the name query from the external ISP.

A service's public name mapping may reveal to malicious core ISPs that they are hosting a rendezvous for that service, but we defend against this by returning a different rendezvous mailbox for each query. Services that do not need this functionality can simply set up a DoS-resilient number of mailboxes and return them repeatedly in a random order. Services that do need to hide their rendezvous mailbox usage must continually set up rendezvous mailboxes to handle new connections. While this means ISPs cannot cache name records, they can be cached at end hosts.

ISP Lists: The other category of information that authorities provide is a list of ISPs and ISP route advertisements. Each ISP chooses a set of authorities to sign its public key, and that public key is hosted by the authority.

The route advertisements explained in §6.1 are also stored by the authorities in order to protect against filtering of advertisements by downstream ISPs. Authorities store route advertisements for all ISPs, not just the ones that they sign. Thus, to block an ISP's routing advertisements, the adversary must block all non-colluding authorities. We note that, in the Internet, there are only about 3000 transit ISP's—ones whose triplets would be needed to construct routes. The ISP lists are thus small and could also be cached for long periods of time since this information changes infrequently.

6.3 Onion Routing in the Network Layer

An end host establishes a circuit to some rendezvous point by setting up an onion-routing circuit to the rendezvous mailbox using the path advertisement triplets described in §6.1 and §6.2.

The end host connects to the first-hop ISP's onion router by encrypting the first half of a Diffie-Hellman handshake, g^a with the first-hop ISP's public key and sending it with a unique hop ID in a packet of the form: $<$*Create, hop ID*, $\{g^a\}_{P(ISP)}>$, where $P(ISP)$ is the public key of the ISP that controls the first onion router.

When the onion router receives the request, it responds along the ingress interface with the second half of the Diffie-Hellman handshake, g^b.

The end host chooses a sequence of path-advertisement triplets that do not necessarily form a shortest path toward the rendezvous mailbox. It is important that not everyone on the circuit be colluding, which becomes difficult when a malicious root authority creates Sybil ISPs and mailboxes. A reasonable heuristic for this choice is requiring that the circuit pass through a few core ISPs as well as a few ISPs whose public key is signed by a different root authority.

Subsequent onion routers are added to the path in a similar way except that the end host uses the last onion router in the path to trampoline to the next one. The end host onion encrypts the *Create* packet with the ISP public key of the new onion router and then the public keys of the sequence of onion routers on the path.

Therefore, cells traveling away from the end host are onion-encrypted with the negotiated symmetric keys of the onion routers and these layers of encryption are peeled away at every hop. Packets traveling toward the end host have another layer of encryption added at each onion router using the negotiated symmetric keys.

The performance penalty for setting up a circuit can be minimized by prefetching path creation. Since all rendezvous mailboxes are in the core, the end host can set up a path to some core ISP, and, when it needs to connect to a particular rendezvous point, set up the final few hops.

7. DISCUSSION

7.1 Safety Properties

Anonymity: Onion routing ensures that, as long as not everyone on the onion routing circuit is malicious, anonymity can be preserved. By ensuring a circuit travels through many ISPs before attempting to access content, we ensure that it is not likely that the entire circuit is colluding with each other. This is done on both ends of the connection, and so a given rendezvous node cannot be tied to either the source or the destination. It is also the case that the source, destination, and rendezvous node are hidden to everyone except the ISPs in which they are located, and that the one-time-use name resolution protects the service of connections.

DoS Resilience: DoS resilience in our system is largely provided by two mechanisms: an inability to send directly to end hosts and a multitude of rendezvous mailboxes.

All traffic to end hosts is indirected through rendezvous mailboxes. Traffic to an end host can only occur if the end host has already set up a path and has not chosen to tear it down. Otherwise, there is no path to the end host and no way to find a route without compromising the end host's first-hop router and sending traffic directly from that router.

DoS attacks on rendezvous mailboxes do not matter as each service has many of them. The only other entities that are DoSable in our system are routers and links. Even though an attacker cannot send directly to any of these entities, a carefully crafted attack can overload a router or link in a core ISP, but we assume that core ISPs are over-provisioned and well-engineered so that they can handle large amounts of load.

Censorship Resistance: We assume that, for any given end host, there exists some ISP that is willing to provide connectivity (but may still seek to censor incoming or outgoing traffic). Our anonymity properties make it difficult for ISPs to block a subset of the packets without cutting off the end host entirely from the outside world. In fact, as long as an end host can get outside of a group of ISPs that censor, it can trampoline to another part of the network or any service.

For a service, we assume that there is some authority that is willing to store the mapping from name to public key and rendezvous nodes. Thus, we can handle cases similar to Wikileaks, where some authority is willing to store the name mapping even though many ISPs and other authorities may be adversarial.

7.2 Path Dilation

Our routing algorithm forces all traffic through the core ISPs in order to reach the rendezvous mailboxes. In order to estimate how much path dilation is caused by this requirement, we examine using IPlane traceroute data[1] what portion of current Internet paths go through core ISPs. If we conservatively assume that the core consists of only 11 Tier 1 ISPs, 60.5% of paths out of 13M paths travel through core ISPs; doubling the number of cores raises the proportion to 76.7%. In addition, the median AS path length of the original path is 5, and only 1% of paths have less than 3 AS hops, the number of hops used by Tor. Thus, we expect the mandatory path dilation for

[1] http://iplane.cs.washington.edu: We used traces from July 14th, 2011.

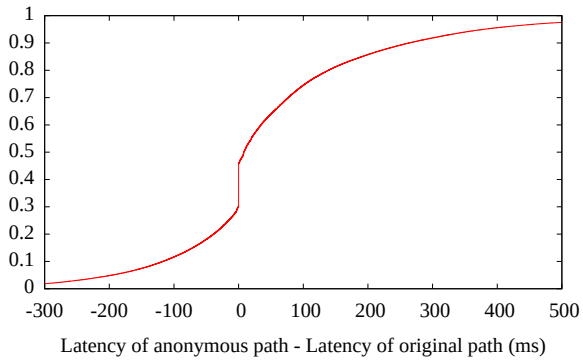

Figure 2: CDF of dilation

end hosts that do not need stronger anonymity guarantees will be small.

For a more detailed analysis, we calculate the difference between a path in our system and the original path. Given a source/destination pair, we use IPlane traces[2] to calculate the latency of an anonymous path by adding the latency between the source and a router in a Tier 1 and the latency between the router and the destination. If there exist multiple such paths, we average latencies of the paths. Figure 2 depicts CDF of the latency differences. Interestingly, paths through cores show even lower latencies than the original paths for about 30% of paths, and 74.5% of paths show less than 100ms of increased latency.

7.3 Economic Incentives

We maintain as much of the Internet's model as possible to ensure feasibility of deployment and alignment with economic incentives. In our system, ISPs still connect to each other through peering or transit relationships that are determined by economic agreements—there is a set of core ISPs that peer with each other, and they make money by charging smaller ISPs for transit. We extend this slightly to allow back-to-back traversal of peering links in the core, but this simply adds some amount of load and does not affect payment systems. Traffic outside of the core remains roughly equivalent to current traffic patterns. Similarly, services find and pay for name resolution services exactly as they do today.

We add two additional components: onion routers and rendezvous mailboxes. Both onion router and mailbox utilization should be closely correlated with bandwidth utilization, thus preserving the pricing and provisioning model of the Internet.

[2]We filtered out traces having end-to-end latencies more than 800ms, or last-hop latencies more than 100ms. Those traces are considered as measurement errors due to the network condition. Also, we performed dilation analysis for a random sample of 450K source-destination pairs.

8. CONCLUSION

We presented a network layer protocol that provides many desirable technical properties: anonymity, censorship resistance, network neutrality, DoS resilience, and decentralization. These properties are all complementary, but especially in the case of anonymity vs. DoS resilience, can sometimes be conflicting at the same time. Our system also attempted to ensure that the interface to higher layers (which includes performance and usability) and the economic incentives of entities in the Internet are minimally disrupted.

Our solution is intended to be a clean-slate design, but can be incrementally deployed as well. Some subset of ISPs can add onion routers and rendezvous mailboxes to the middle of the network. The onion routers of a particular ISP would collectively share one IP so that end hosts and other onion routers could route to them, even if they are not adjacent on the AS graph. Both clients and service hosts can use any onion router as an entry point into the network, and any rendezvous mailbox as an exit point. Services that use the system need to advertise the rendezvous mailbox addresses, perhaps in the same way as CDNs in the current Internet.

Such a partial deployment would only require a few ISPs to participate across the entire Internet and would provide anonymity guarantees, but not any protection against censorship. In order to prevent DoS attacks, it is necessary to disallow any traffic through the current Internet and so end hosts looking for DoS protection must have participating first-hop ISPs and receive all traffic through rendezvous mailboxes. We would also, depending on the underlying BGP policies, lose policy-compliance for the end-to-end path with this scheme as a single flow of traffic could travel through valleys. However, this would still constitute a good first step toward baking support for our desired properties into the network.

9. REFERENCES

[1] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Dns security introduction and requirements. RFC4033, 2005.
[2] R. Dingledine, N. Mathewson, and P. Syerson. Tor: The second-generation onion router. In *USENIX Security Symposium*, 2004.
[3] C. Dixon, T. Anderson, and A. Krishnamurthy. Phalanx: Withstanding multimillion-node botnets. In *NSDI*, 2008.
[4] P. B. Godfrey, I. Ganichev, S. Shenker, and I. Stoica. Pathlet routing. In *SIGCOMM*, 2009.
[5] S. Kent, C. Lynn, and K. Seo. Secure border gateway protocol (s-bgp). *IEEE Journal on Selected Areas in Communications*, 2000.
[6] K. Lakshminarayanan, D. Adkins, A. Perrig, and I. Stoica. Taming IP packet flooding attacks. In *HOTNETS*, 2003.
[7] X. Liu, A. Li, X. Yang, and D. Wetherall. Passport: secure and adoptable source authentication. In *NSDI*, 2008.
[8] B. Raghavan, T. Kohno, A. C. Snoeren, and D. Wetherall. Enlisting isps to improve online privacy: IP address mixing by default. In *International Symposium on Privacy Enhancing Technologies*, 2009.
[9] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet indirection infrastructure. In *SIGCOMM*, 2002.
[10] X. Zhang, H.-C. Hsiao, G. Hasker, H. Chan, A. Perrig, and D. G. Andersen. SCION: Scalability, control, and isolation on next-generation networks. In *IEEE Symposium on Security and Privacy (Oakland)*, May 2011.