Tariq Elahi, Colleen M. Swanson, and Ian Goldberg

Slipping Past the Cordon: A Systematization of Internet Censorship Resistance

Abstract: The Internet censorship conflict is like a game of cat-and-mouse, in which the censor discovers and acts on *distinguishers* that set the usage of censorship resistance systems (CRSs) apart from other Internet usage, and the CRSs' attempts to remove and mitigate these distinguishers. Our systematization is based on a simple and intuitive model of the distinguisher-leveraging censorship apparatus, but one that is grounded in the inherent limitations of the censor and the unavoidable error rates of the apparatus used. We are then able to categorize CRS techniques and designs in terms of their impact on distinguishers, and thereby evaluate a broad range of systems with respect to their security and applicability for certain use cases. We present a taxonomy of censorship circumvention strategies and techniques and an evaluation of the systems that utilize them. Our evaluation leads us to identify gaps in the literature, question the assumptions at play, and formulate a plan of action to develop effective CRSs.

1 Introduction

The Internet is a valuable tool for political freedom; the ease with which information can be disseminated has been a boon to creating social change. For example, the history of the past few years shows that the events of the Arab Spring were in part spurred by the ability of revolutionaries to mobilize and organize the population through the use of social networking tools [7, 27]. However, nation states (as well as other actors) regularly curtail or control access to the Internet, thereby chilling speech and otherwise limiting the free flow of knowledge. As a result, numerous *censorship resistance systems (CRSs)* have been developed with the aim of providing access to blocked content, anonymous publishing and browsing services, and secure, private communications.

The Internet censorship conflict is like a game of cat-and-mouse: as the censor learns how to be more effective, the circumventor learns how to overcome, which in turn informs the censor. This cycle repeats until one side reaches the limits of the resources they are willing and able to invest and the costs they are willing to bear, at which point a tentative equilibrium is reached. The resulting balance, however, may later shift as technology, policy, and resource availability changes.

While researchers have made great advances in providing circumvention solutions, there is a great need for a systematization of knowledge, as we currently lack a common framework for the design and evaluation of CRSs. Often the confusion between intrinsic properties (those that the system can control through design) and extrinsic properties (those that emerge from the environment and are not controlled by the design) clouds matters when we wish to evaluate a CRS. A formal framework would allow us to separate the intrinsic properties that are critical for effective CRSs and the extrinsic properties that prevail in the operating environment. We do not yet know how to comprehensively evaluate CRSs, including their ease of deployment, difficulty of detection and blocking, effectiveness under different censorship scenarios, and how they compare with one another.

In this paper, we categorize and model the behavior of the censor, observing that the censor must act on *distinguishers*, or features of the CRS that set it apart from other Internet services. Specifically, censor activities can be understood in three phases: learning, detection, and response. The effectiveness of a censor's response mechanism may be specified in terms of its false positive and false negative rate. We then provide a qualitative evaluation of CRS strategies, which aim to hide or avoid the use of distinguishers, and the impact of these techniques on censor error rates, and present a comparative analysis of state-of-the-art CRSs. By careful consideration of the current design landscape, we identify gaps in current research, including obstacles to deployability, and the need to reassess common design assumptions with respect to the censor. In particular, current CR research is rooted in three main assumptions, which we argue merit further consideration: the effectiveness of collateral damage strategies; the ability of the censor to block traffic based on sophisticated traffic analysis; and the image of the censor as an entity with a limited *sphere of influence* and *sphere of visibility*, whose main goal is to prevent undesirable communication, either directly using network-level blocks or indirectly through chilling techniques such as surveillance.

An outline of our paper is as follows. In section 2, we discuss the censor and give a model of a generic censorship apparatus. We give an overview of censorship resistance systems, including their common components and desirable security and performance properties in section 3, and examine

Tariq Elahi: University of Waterloo, E-mail: tariq.elahi@uwaterloo.ca
Colleen M. Swanson: University of Michigan, E-mail: cmswnsn@umich.edu
Ian Goldberg: University of Waterloo, E-mail: iang@cs.uwaterloo.ca

common attacks on such systems in section 4. In section 5 we examine the CRS space, discussing common censorship resistance strategies and their limitations, and apply them to existing CRS designs. We also identify trends and look ahead in section 6, consider related work in section 7, and finally conclude with a call for action in section 8.

2 Internet Censorship

Censors vary widely with respect to their motivation, sphere of influence, and technical sophistication. A wide range of entities, from individuals to corporations to state-level actors, may function as a censor. Therefore, a CRS should be precise as to the type and capabilities of the censor(s) that it is designed to circumvent.

The extent to which a censor can effectively attack a given CRS is a consequence of that censor's resources and constraints. We will focus on the censor's technical resources, capabilities, and goals which are informed by their *sphere of influence* (SoI), their *sphere of visiblity* (SoV), and their set of economic constraints. The sphere of influence (SoI) is the degree of *active* control the censor has over the flow of information and behavior of individuals and/or larger entities. The sphere of visibility (SoV) is the degree of *passive* visibility a censor has over the flow of information on its networks and those of other operators. The success or failure of a CRS depends in large part on the assumptions its designers make about the SoI and SoV of the censor.

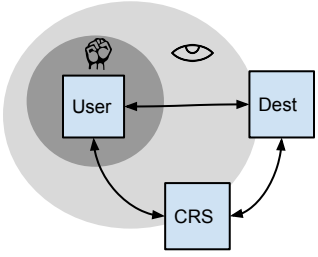

Fig. 1. An example SoI (dark gray area with fist) and SoV (light gray area with eye); arrows depict communication channels between entities. The user-destination channel falls within both the SoI and SoV; the destination is outside the SoI, but may be within the SoV.

Figure 1 depicts a common SoI and SoV scenario. The user is within the censor's SoI and SoV, as is all of his outgoing traffic, including to the CRS. Even though the CRS is outside the SoI, the censor may be able to influence or block user–CRS traffic. A well-designed CRS prevents this, often by ensuring the censor would have unacceptable collateral damage (i.e., would block high-value innocent traffic). Similarly, the censor can block all Internet traffic within its SoI, but we note that real-life examples of this have been short-lived. [14, 54]

This is the essential challenge of CRS design: to allow users to access censored content and destinations while within the influence and visibility of the censor, while keeping the censor none the wiser.

The limitations of the SoI/SoV may be *physical*, *political*, or *economic*. Limitations due to geography are an example of physical constraints, while relevant legal doctrine, or international agreements and understandings, which may limit the types of behavior in which a censor is legally allowed or willing to engage, are examples of political limitations. Economic constraints assume the censor operates within some specified budget which affects the technical sophistication and accuracy of the censorship apparatus it can field.

2.1 Censor Threat Model

We now define our threat model based on the censor's network-level capabilities. A censor has complete visibility of network traffic flows within its SoI. It may have additional visibility outside of its SoI (but within its SoV), such as information gained from collusion with third parties. In particular, the censor may be a *passive* adversary with respect to some portion of the network, only able to observe channel content, such as when an ISP grants access to past network traffic flows.

The censor may be *active* with respect to some portion of the network (within the SoI), with the ability to *modify*, *delete*, *inject*, and *delay* channel content. Furthermore, the censor may have the ability to modify channel *characteristics* as well as content, such as controlling the timing patterns of traffic. The censor may have additional insight into the CRS system that may be gleaned by being an active CRS participant or by devoting resources in a volunteer-based CRS.

The censor has two goals. First, the censor wishes to *detect and identify* CRS traffic; the censor wants to be able to detect individual CRS traffic flows and may also wish to expose the identity of the user or the content of the material accessed. Second, the censor wants to *disrupt* traffic and render the CRS ineffective.

2.2 A Censorship Apparatus Model

At an abstract level, the censorship apparatus is composed of classifier and cost functions that feed into a decision function (see Figure 2). Censorship activity can be categorized into three distinct phases: *exposure*, *detection*, and *response*. We will leverage this model in our censorship resistance taxonomy in section 5.

In the first phase, the censor exposes a set of distinguishers $\mathcal{D}$ that can be used to identify prohibited network activity within some acceptable margin of error. The classifier, once deployed, takes $\mathcal{D}$ and the set of network traffic to be analyzed,

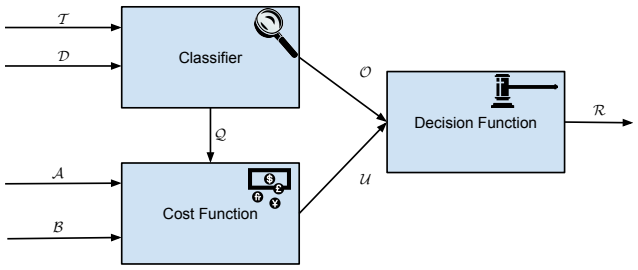

Fig. 2. Censorship apparatus model. The classifier takes as inputs the set of network traffic to be analyzed, $\mathcal{T}$, and the set of distinguishers, $\mathcal{D}$, and outputs a set $\mathcal{O}$ of traffic labels and the associated FNR and FPR, denoted by $\mathcal{Q}$. The cost function takes as inputs $\mathcal{Q}$, together with the censor's tolerance for collateral damage (FPR) and information leakage (FNR), denoted by $\mathcal{A}$ and $\mathcal{B}$, respectively, and outputs a utility function, $\mathcal{U}$. The decision function takes $\mathcal{O}$ and $\mathcal{U}$ as inputs and outputs a response $\mathcal{R}$.

$\mathcal{T}$, as inputs. As an example, a censor may learn that a particular packet length is dominant for most, but not all, of a given CRS's traffic, say the 586-byte length prevalent in Tor. There is also non-CRS traffic that has this particular packet length that the classifier may misidentify; the censor must determine an appropriate occurrence threshold and if the resulting error rates are acceptable before using this pattern as a distinguisher.

The censor's error rates are the *information leakage*, or *false negative rate (FNR)*, which is the rate at which CRS traffic is mislabeled as legitimate, and the *false positive rate (FPR)*, which is the rate at which legitimate traffic is mislabeled as CRS-related, thus causing *collateral damage*, or the blocking of legitimate traffic. We assume the censor prefers to minimize the FNR and FPR within its given set of constraints.

In the second phase, the classifier *detects* and flags suspicious network flows using a classification mechanism. In our example, the classifier may flag flows that exhibit 586-byte packet lengths more often than some censor-defined threshold.

In the third phase, the censor *responds* to flagged network flows, relying on a utility function that accounts for the censor's costs and tolerance for errors. The response also depends on the actual mechanism employed by the CRS and its susceptibility to interference. In our example, the censor may choose to block flagged network flows by sending RST packets to the server.

Distinguishers play a vital role for the censor. Formally, a distinguisher is composed of feature and value pairs. A *feature* is an attribute, such as an IP address, protocol signature, or packet size distribution, with an associated *value* that can be a singleton, list, or a range. In general, values are specified by a *distribution* on the set of all possible values that feature can take. Where this distribution is sufficiently unique, feature-value pairs can be used as a distinguisher to detect prohibited activities. In our example above, a prevalence of 586-byte packet lengths forms a distinguisher the censor can utilize to identify Tor traffic.

The primary source of distinguishers is network traffic within the censor's SoV; we can map these distinguishers to the network, transport, and application layers of the network stack. Distinguishers can be extracted from the feature-value pairs within headers and payloads of protocols at the various network layers, e.g., source and destination addresses in IP headers, source and destination ports and sequence numbers in the TCP header, and TLS record content type in the TLS header. Packet payloads, if unencrypted, can also reveal forbidden keywords; the censor may act on such distinguishers without fear of collateral damage since there is little uncertainty about the labeling of these flows.

The censor can use the explicit distinguishers above to learn about the expected behavior of certain types of traffic. A certain profile of feature-value pairs implies particular, and known, traffic behavioral characteristics that the censor can use to detect anomalous, and potentially CRS, traffic. For instance, a VoIP channel usually has data flowing in both directions but rarely at the same time, as both parties are usually communicating interactively and taking turns speaking. A CRS that uses this channel for bulk download would only have data flowing predominantly in one direction, with some data flowing back for data transmission control; this discrepancy can be used as a distinguisher. To augment these explicit distinguishers, the censor can collect traffic statistics to determine additional distinguishers, such as packet length and timing distributions.

Distinguishers are high- or low-quality depending on if they admit low or high error rates, respectively. Furthermore, they can be low- or high-cost depending on if little or large amounts of resources are required to deploy them. For the censor, high-quality, low-cost distinguishers are ideal, whereas the CRS attempts to avoid having any distinguishers, but in practice relies on forcing the censor to depend on only high-cost, low-quality distinguishers.

3 Censorship Resistance

Censorship resistance is the art of successfully overcoming the censor's SoI and SoV while achieving an acceptable level of security and performance for its participants. In this section we distill the common components of CRSs, the desirable user and system security and performance properties, and the resultant overall CRS design goals.

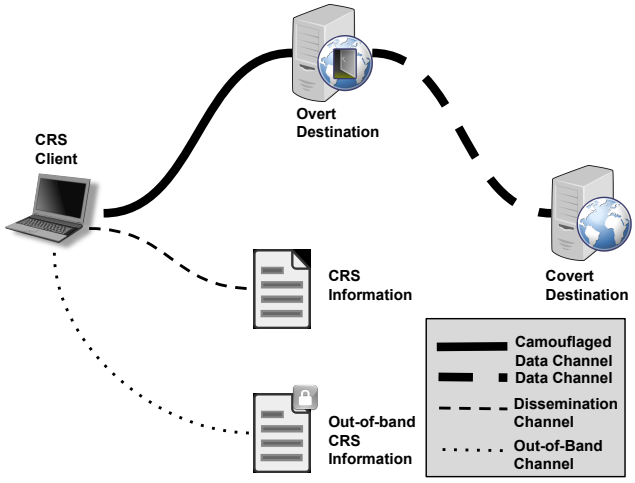

Fig. 3. High-level components of a CRS. A user learns how to access the CRS and other relevant information through the *dissemination channel*, an *out-of-band channel*, or both. The user can then connect over the *data channel* to an unblocked *overt destination* that acts as a CRS proxy; this connection may or may not be camouflaged to prevent detection. The proxy then services the user's request, either by fulfilling the requests locally or by forwarding the request to one or more *covert destinations*.

3.1 Censorship Resistance Components

Figure 3 depicts the basic components of a CRS which we now describe in detail:

Covert Destination. This is the covert page, person, or platform to which the CRS client ultimately wants access; this definition allows for use cases that require interactivity as well as read and publish mechanics. Examples include a dissident blog (page), a video call with a friend (person), and tweeting about the location of the next anti-government rally on Twitter (platform).

Overt Destination. This is the client's ostensible destination from the viewpoint of the censor; its exact role varies across CRS designs. It can behave as an active *forwarder* when the client wants to communicate with or gain access to censored destinations that are external to the CRS, by proxying traffic to and from the client and external destination. It can also store and serve the censored content locally, acting as a *covert destination/data store*. Finally, it can be a passive *dummy destination*, for example if the CRS leverages networking equipment to divert the CRS traffic away from the dummy destination and towards the covert destination.

Data Channel. There are two types of data channels, which may use different network mechanisms and have very different security and performance properties: one between the client and overt destination, and another between the overt and covert destinations; not every CRS uses the latter. These channels transport censored data to and from the client and the overt destination, and where needed to the covert destination. The client-to-overt-destination data channel may be camouflaged if the censor uses traffic analysis to detect CRS activity. The overt-to-covert-destination data channel, if one exists, may or may not be camouflaged, as it is outside the censor's SoV and, generally, SoI.

CRS Information. In general this is any information that the client needs to participate in the CRS, such as addressing, naming, routing, and authentication details. Some of this information may be public, such as domain name mappings to IP addresses, routing information, and other general Internet communications details. CRS-specific information is usually also required, including addressing information of proxy servers or locations of censored content; under certain designs this CRS-specific information is restricted to prevent the censor from learning or tampering with it.

Dissemination Channel. CRS information is requested and served over a dissemination channel, which is within the censor's SoV and often SoI. This channel can take many forms, such as ordinary email to a publicly available address, a CRS-client-issued lookup request to a directory serving a file, or an anonymized encrypted connection to a hidden database. Generally, the throughput and network characteristics of the dissemination channel are more limited than the data channel.

Out-of-band CRS Information. Certain CRSs require secret information to *bootstrap* the data and dissemination channels. This may include setup or operational information such as symmertric keys, network dead-drop locations, and any other information that is necessary for the client to learn how to participate in the CRS.

Out-of-Band Channel. A channel outside the censor's SoI and SoV is termed an *out-of-band channel*. This channel is used to communicate CRS-operational secrets that are assumed to somehow arrive with absolute security with respect to the censor. The key is that there is a limitation inherent to the channel that does not allow it to be used as an ongoing data or dissemination channel, which would also obviate the need to use a CRS in the first place. An example of such a channel is a person from outside the SoV bringing addresses of CRS proxies on a USB stick through the airport and hand delivering it to the client.

CRS Client. This is special client software that uses CRS information, channels, and destinations to provide the desired CRS functionality to the client. A client may need to obtain this software through an out-of-band or dissemination channel.

3.2 CRS Users and Use Cases

Users of CRSs include whistleblowers, content publishers, citizens wishing to organize, and many more, and any one CRS

may be employed simultaneously for multiple use cases. Each use case may require different properties of the CRS. For example, a whistleblower needs to have their identity protected and be able to get their message to an outsider without raising suspicions that this is occurring; a content publisher wants that no amount of coercion, of themselves or the hosts of their content, could impact the availability of their content; and citizens wishing to organize would want unblockable access to their self-organization tool.

We consolidate from these various use cases a coherent set of security and performance properties, appearing in boldface below, with the aim that they remain applicable to the use cases from which they stem, keeping in mind that not every CRS design aims to achieve every property. This use-case based consolidation provides consistent evaluation criteria applicable across all the different types of CRS designs we survey and allows us to focus on their user impact, i.e. the use cases enabled, rather than their technical details. We describe the security properties next.

An **unblockable data channel** is one that the censor, having identified it, is unable to block due to extenuating circumstances, most usually the threat of collateral damage. If the censor is unable, or unwilling, to block the data channel then the CRS activity can proceed unmolested.

An **undetectable data channel** is one that the censor is unable to identify as belonging to CRS activity. This is a useful for cases where the censor *could* block the channel if they knew it were CRS related.

An **anonymous publisher** is one whose identity is hidden from CRS-participating entities and observers on the Internet in general. This is to prevent the chilling of speech that the threat of retribution by the censor would cause on the publisher.

An **unlinkable client-destination path** is one where an observer, within the CRS network or outside it, cannot tell which destination a particular client is communicating with. This prevents the censor, or any other observer, from learning whether the client is communicating with known undesirable, and censored, destinations.

An **incoercible publisher** is one for whom the censor's threats of force are not credible due either to the fact that the publisher is outside the censor's reach or to the fact that the CRS is designed in such a way that makes it impossible to comply with the censor's demands.

An **incoercible covert destination/storage** is one for which the censor's threats of force are not credible for the same reasons as those for the incoercible publisher above.

A **deniable client participant** is one whose participation in the CRS is blurred with innocuous activities to create reasonable doubt in the censor's mind and give the client plausible reasons for their, secretly CRS related, activities. This is to mitigate the threat of censor retribution and the chilling of speech.

A **deniable forwarder participant** is, like the client participant above, one whose participation is covered in reasonable doubt and this helps to prevent the censor from blocking it or meting out punishment.

A **deniable covert destination/storage participant** is, like the forwarder participant above, one whose participation is covered in reasonable doubt for the same reasons.

Along with the requirements for security, CRS clients' usage is impacted by what is possible to accomplish given the performance properties of the communication channel, which we define next.

Latency is the delay introduced by the CRS in carrying out a user action, e.g. sending a message, or receiving a file. The two types of CRS are low-latency and high-latency with the former being useful where delays would impact usage, e.g. web browsing, and the latter for uses that are not impacted by delays, e.g. leaking a document to a whistleblowing website.

Throughput is the amount of bandwidth that the CRS needs to harness for both its own functionality and to enable the use cases it was designed for. There are two types: high-throughput and low-throughput. An example high-throughput CRS is Tor where it consumes the bandwidth dedicated by numerous volunteer routers that form the network, while Collage is a low-throughput CRS since it can only harness bandwidth on the same order of magnitude as the size of image files that are posted on photo-sharing websites.

Goodput is the useful bandwidth available, of the total throughput bandwidth above minus the CRS's operational overhead, for CRS user activity. A high-goodput CRS means that the CRS is high-throughput and has low operational overhead resulting in high amounts of bandwidth available for user activities. A low-goodput CRS means that either the CRS is low-throughput to begin with or that the CRS is high-throughput but has high overhead resulting in little bandwidth for user activities. In the CRS examples above, Tor is a high-goodput CRS and is useful for large data transfers, while Collage is low-goodput since it uses steganography, that adds a large overhead but is still useful for smaller data-footprint activities.

Reachability is the range, or extent, of the Internet—which is composed of people, web pages, and service platforms—that a CRS user can communicate and/or interact with. The range can be general or limited; general being the same as unfettered access to the Internet and limited being a curtailed set of niche destinations, audiences, and/or content. The former is useful where the user

wants transparent access to the Internet such as a user who wishes to instant message with friends, watch streaming movies, and post to social networking websites. The latter is useful where the user only requires a limited part of the Internet such as communicating amongst a tight-knit group of dissidents who only require access to one bulletin board system.

Interactivity of the CRS channel tells us whether bidirectional communication is possible between CRS users or whether it is only unidirectional. A CRS that is bidirectional means that it can be used by a user to both send and receive data, whereas a unidirectional CRS can only transmit or receive data during any given session of use of the system. A unidirectional CRS is useful where the user only wishes to post to a message board but will not read the board themselves or when they only wish to access information but will not be adding anything themselves. A bidirectional CRS is useful for times where users needs to interact with, and react to, other users and there is a need for a back and forth within the same session of use of the CRS.

These channel performance properties, with their high/low, general/limited, or bi-/unidirectional settings, define the *quality of service*[1] (QoS) that a CRS user can expect. The channel QoS dictates the user experience and the kinds of activities that are possible on that channel and hence the use cases that it can support. We will see later in subsection 5.5 how well CRS designs address these user requirements for security and QoS.

3.3 Desired CRS Design Goals

We are now in a position to consolidate what we have learned into a coherent set of CRS functionality, security, and performance design goals.

At the very general level a CRS has the following functionality goals:

- *CRS Information Discovery*: The user must be able to learn proxy and/or content network locations, as well as auxiliary information required to participate in the CRS.
- *CRS Data Transport*: The CRS data channel must be able to punch through the censor's technological barriers and connect users with destinations (pages, people, platforms) from which they would otherwise be blocked.
- *CRS Bootstrapping and Availability*: The CRS should be designed to deploy on the Internet of today and maintain consistent availability. It should not depend on features and functionality that have not yet become, and are not likely to become in the near term, a reality on the Internet.

A CRS must ensure the security of the above three functions as well as that of CRS users and operators. We now consolidate the required CRS-security properties, that we have extracted from the literature on attack surfaces and vectors, and real-world examples. They are organized by the three phases of censorship from our model in subsection 2.2:

Exposure Mitigation

- *Prevent CRS information harvesting* that is possible by masquerading as a CRS client and combing the Internet address space.
- *Prevent CRS participant identification* that is possible by masquerading as a CRS client or overt/covert destination.
- *Prevent active probing* by the censor who is trying to distinguish whether CRS infrastructure or content is present at a particular network location.

Detection Mitigation

- *Provide an undetectable data channel* that camouflages traffic patterns, behaviors, and inconsistencies detectable by the censor.

Response Mitigation

- *Provide an unblockable data channel* that operates in a curtailed networking environment that cannot be blocked without collateral damage to non-CRS traffic.
- *Provide user and operator anonymity and deniability* to prevent chilling effects and self-censorship as well as to mitigate coercion.
- *Provide resilience* against the censor actively participating in, compromising parts of, removing parts of, and adding malicious resources to the CRS.

Finally we address the desired performance properties, discussed in subsection 3.2, and note that these are driven by specific use cases. However, it is clear that if all the properties are maximized, i.e. provide the highest level of performance, then all use cases will be covered.

Therefore, the desired *ideal* CRS is one that provides all of the security properties and has **low latency**, **high throughput** and **goodput**, has **general reachability**, and allows **bidirectional communication**. However, in reality we see that there are few CRS designs that provide all of these properties; often there is a trade-off between security and performance. We shall see evidence of this later in Table 2 where we summarize security and performance properties of real-world CRS designs. Thus, the types of use cases is driven by the particular security and performance properties that the CRS achieves.

1 Usually quality of service is concerned with the performance of a given channel (i.e. latency, throughput, and goodput); we add CRS-specific channel characteristics (reachability and interactivity) since these are also pertinent in the CRS setting.

4 Attacks on CRS Components

Each of the CRS components within the censor's SoI or SoV is a potential attack surface; we will survey each to gain insight about the particular issues and challenges inherent to each component.

4.1 CRS Information

Harvest: The censor can readily harvest any public CRS information, and where it is somehow restricted, the censor can actively participate in the network, or otherwise infiltrate or compromise the CRS. This is exactly the manner in which some governments, such as China [57], have taken to block access to the Tor network.

Poison: The censor can modify public information and inject or modify CRS traffic within its SoI. Examples include DNS poisoning [4] and routing table changes that isolate traffic to known, trusted, and censored paths [69]. The censor needs to be careful when corrupting information [44] needed for Internet operations as it can impact non-CRS use. [64]

4.2 Dissemination channel

Deny: The dissemination channel may be blocked to prevent clients from accessing CRS information.

Masquerade: If the CRS depends on secrecy, the censor can pretend to be a legitimate client and use the dissemination channel to harvest CRS information, as mentioned above. China has twice overwhelmed Tor's bridge distribution strategies by pretending to be legitimate users from different subnets on the Internet [16].

4.3 Data channel

Detect: From unencrypted header information, the censor can tell if traffic is going to a known CRS overt destination. The content (or payload) may also provide evidence that the packet is CRS-related; if this information is unavailable or obfuscated, traffic analysis based on packet statistics and behavioral signatures can be used. The censor can compare these clues against known CRS behavioral patterns. [15] Where the clues match known non-CRS statistics and behaviors, the censor can look for any discrepancies that give away the pretense.

Deny: The censor can drop all identified CRS traffic at the firewall or another point of presence, effectively severing the CRS data channel. For example, the Chinese firewall, circa 2004, sent RST packets to both ends of a network flow whenever it detected a Tor connection. [13]

Disrupt/Degrade: Alternatively, the censor can manipulate traffic by injecting spurious packets, dropping key portions of the traffic, modifying the contents of packets [59], or otherwise manipulating the characteristics of the underlying network link, e.g., by introducing delays, low connection timeout values [73], and other constraints. These techniques are especially useful if the censor is unsure whether the suspicious flow is CRS-related, and can be very effective if the CRS data channel is brittle and not resilient to errors, but legitimate, non-CRS traffic, is not similarly degraded. [34, 60] The censor can leverage discrepancies between CRS and non-CRS data-channel error handling to degrade the CRS while leaving the non-CRS data effectively unmolested. These tactics can guard against collateral damage that would otherwise be incurred if the censor were to employ more drastic measures like severing network flows.

4.4 Overt and Covert Destinations

Comb: The censor may comb the Internet, probing destinations for telltale signs of CRS activity. For example, China has been very aggressive in probing for bridges by watching outbound TLS connections to U.S. IP ranges and then following up with Tor connection requests to verify the existence of a bridge. After this confirmation step, all TLS connections to identified bridges are dropped by the Chinese firewall. [80]

Coerce: Within its SoI the censor can use legal or extralegal coercion to shut down an overt destination. History shows that this can been successful, e.g., a popular anonymous email service was pressured by the U.S. government to reveal users' private information [33].

Crush: For destinations outside the censor's SoI, the censor can mount a network attack, e.g., China launched an effective distributed denial of service attack (DDoS) against Github, to target censorship resistance content hosted there. [37] Other attacks based on infiltration of resource pools can cause similar denial of service.

Curtail: A censor can easily block IP addresses of overt destinations newly deemed undesirable with firewall filter rules and DNS poisoning or injection attacks. Such methods are widespread; Aryan et al. [2], Nabi [66], and Clayton et al. [13] provide evidence of this for Iran, Pakistan, and China, respectively. The censor can also use a *whitelist* to allow access to approved destinations only.

Corrupt: The censor can set up malicious resources, such as proxy nodes or fraudulent documents that counteract the CRS's goals, to attract unwary CRS clients and negatively impact their CRS usage. For example, adversarial guard relays are known to exist on the Tor network and can be used to compromise Tor's client-destination unlinkability property. See Elahi et al. [25] and several follow-on works [6, 19, 48] for an in-depth analysis.

4.5 CRS Client

Link: The censor can try to implicate clients for using a CRS, since such systems may be regarded as having only illegiti-

mate uses. The censor can do this by identifying client connections to known CRS-participating overt destinations. The censor may further be interested in identifying covert destinations and/or content and can do this by attracting clients to use or access censor-controlled CRS resources; again Elahi et al. [25] and the follow-on works [6, 19, 48] provide an investigation of this attack vector.

Coerce: The censor can try to pressure publishers, e.g. through threats of imprisonment, into retracting publications and otherwise chill their speech. China regularly regulates the online expression of its citizens, using an army of thousands of workers to monitor all forms of public communication and to identify dissidents [87], who may then be targeted for punishment.

Compromise & Deny: The censor can install monitoring software on client computers. This may be publicly announced, such as in the case of Green Dam Youth Escort [83], and the censor may impose rules demanding clients install only compromised communication software, e.g., TOM-Skype [63], or the censor may install malware covertly. The censor can enforce these rules at the client level by disallowing unapproved software installation, disrupting functionality of Internet searches by returning pruned results, and displaying warnings to the user to dissuade them from attempting to seek, distribute, or use censored content.

5 Examining the CRS Space

Our aim is to better understand how a CRS and censorship apparatus interact. To that end we have constructed a generic censorship apparatus model in subsection 2.2, described the common CRS components in subsection 3.1, distilled the salient security and performance properties from the literature in subsection 3.3, and outlined the major common attack vectors in section 4. We shall now categorize mitigation strategies adopted by CRS designs and analyze their interaction with the error rates inherent in the detection phase and the upward pressure these strategies apply on these error rates. We will then discuss the limitations of these strategies and round out the discussion in this section by applying our models and taxonomy on the CRS designs in the literature.

Figure 4 is a graph of the FNR vs. FPR of a generic censorship classifier. Each error rate lies in the range $[0, 1]$; the origin represents an ideal classifier and the diagonal line joining the points $(0, 1)$ and $(1, 0)$ represents a completely ineffective classifier doing random guessing—point $(p, 1 - p)$ represents a system in which the censor reports a positive classification with probability p, independent of the input. This graph bears similarity to the more common receiver operating characteristic (ROC) curve; the only change has been to replace the true positive rate (TPR) with its complement, FNR. The actual classifier is likely to lie somewhere between these two extremes. With the choice of classification mechanism and distinguishers, the censor determines the level of error they are able to tolerate and achieve, given current resource constraints.

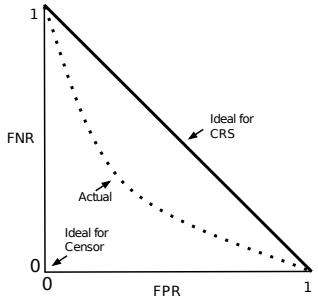

Fig. 4. Tying strategies to the censor FNR/FPR error model. The diagonal line indicates the ideal scenario for the CRS, whereas the origin is the ideal scenario for the censor. The dotted curve shows what a classifier error rate might look like in practice.

CRS strategies aim to push the curve closer to the ideal diagonal (upwards and to the right), and so we say such strategies exert *upward pressure* on FNR/FPR. Of course, with the censor continually applying new techniques to lower rates as the CRS exerts this pressure, the net effect of CRS mitigation strategies may be to keep the curve in the same place or to slow its descent.

We next categorize CRS strategies by the relevant censorship phase and describe how each strategy impacts the censor's error rates and mitigates the attacks outlined in section 4. Table 1 provides a summary and structure for the discussion that follows.

5.1 Exposure Phase

In designs in which CRS information is sensitive, CR strategies often aim to prevent or slow the censor from learning high-quality distinguishers. The main threat is to the dissemination channel and the CRS information itself. The main challenge is that legitimate users must be able to learn CRS information easily, while the censor should not be able harvest efficiently. In this phase, secret information has already been transmitted out of band and initial client setup is complete.

5.1.1 Unpredictable Values (UV)

An unpredictable selection of distinguisher values, chosen from the possible value space that includes non-CRS elements, ensures that the censor cannot implement an *a priori* block of those values. Since the censor has yet to discover distinguisher values, there is a window of opportunity where misclassification can occur, thereby temporarily maintaining the FNR.

One method of generating unpredictable values is to recruit CRS participants outside the SoI who control particular values of a feature, e.g. IP addresses, and use them in a

Table 1. Strategies and techniques employed to defend against censorship threats to CRS components. The CRS components we described in subsection 3.1 and the attacks on each, as discussed in section 4, are organized along the top. The rows are grouped by the phases of censorship described in subsection 2.2. Each group is organized by the strategies that apply to that phase and the techniques that realize them, as discussed in this section. A solid black square indicates that this technique has been applied to this attack vector.

			CRS Component																
			CRS Info.		Diss. Chan.		Data Channel			Overt/Covert Destination					CRS Client		
Phase (Strategy)	Technique	Attack	Harvest	Poison	Deny	Masquerade	Detect	Deny	Disrupt & Degrade	Comb	Coerce	Crush	Curtail	Corrupt	Compromise & Deny	Coerce	Link
Expose (UV)	Actual Value												■				
	Indirect Value									■	■		■				■
Expose (RL)	Time		■										■				
	Value-Space		■										■				
	Proof-of-Work		■			■							■				
	Trust		■			■							■				■
Expose (LI)	Client Authentication					■				■							
Detect (OV)	Traffic Patterns						■										■
	Covert Destination/Content						■										■
Detect (IV)	Adopt						■										■
	Co-opt						■										■
Respond (CV)	Resource-Driven		■		■			■		■			■				
	CRS-Driven		■		■			■		■			■				
Respond (OSoI)	CRS-Specific							■			■					■	
	CRS-Agnostic				■			■			■			■		■	

seemingly random manner. Another method—where the value space is not restricted—is to adjust the distribution of CRS values so that there is no observable pattern. For example, the CRS can adjust the distribution of packet lengths through padding, so that the new distribution has no similarity to the original. In either method, the CRS can use values directly, in which case a censor that learns these values also learns any private information contained therein, e.g. true identity or location, or the CRS can use obfuscated, indirect values, so that even in case of compromise, the censor may not learn the true identity or location, e.g. an email addressed to a pseudonym.

The direct variant is useful for introducing unpredictable forwarder IP address values for CRS usage. FlashProxy [30] runs Javascript proxy code within the browser of any website client who visits a CRS-supporting website and makes them a piece of the CRS. It is irrelevant whether or not this CRS-supporting site is blocked by the censor since these temporary forwarders are created *on the client side* by anyone outside the SoI who visits the site. Tor [21] widely recruits volunteers to operate Tor bridge nodes [20], which are relays whose addressing information is not shared like ordinary Tor relays. The indirect variant is useful for protecting the content location (storage) and covert destination information and users wishing to remain anonymous. Utilizing the Remailer concept based on Chaum's ideas [11], Goldberg and Wagner's Rewebber proposal [36] uses multi-hop routing for encrypted publication on servers whose identities are hidden behind a pseudonymous name space. This makes it difficult for the censor to block

covert destinations since it is difficult to track down the host servers. Furthermore, the server itself is designed to have plausible deniability, as all hosted content is encrypted; a server within the censor's SoI may be able to claim ignorance of the actual content. Tor Hidden Services [21] also use indirection to provide anonymous hosting services.

5.1.2 Rate Limited (RL)

To mitigate a censor that masquerades as a legitimate CRS participant in order to harvest information, the CRS can limit the rate at which the censor can learn feature-value pairs. This is usually done by requiring proofs of work, partitioning the value space over time slices, partitioning the value space over CRS participant attribute(s), or including reputation-based mechanisms. Such strategies work in concert with the unpredictable values technique, ensuring there exist distinguisher values that are not yet known to the censor.

Proofs of work include captchas and other puzzles that require the participant to devote resources to gain information, e.g. Köpsell and Hillig [53] present a puzzle-based challenge when a proxy address is requested, which is designed to be too expensive for wide-scale harvesting by a resource-bound censor. Value-space partitions over time slices ensure that even if harvesting is efficient, the censor can only learn values according to the CRS's timetable, such as in Tor, which releases only a few Tor bridge address at a time. Similarly, partitioning over participant attribute(s) ensures that participants learn disjoint sets over the value space and hence no one participant, who is restricted in resources, e.g. limited IP addresses or low connectivity in a social graph, can learn all CRS information. Examples include distribution of proxy addresses in Feamster et al. [29] and Tor bridge addresses, both of which rely on user IP addresses. In reputation-based schemes, a behavioral component is used to limit the damage caused by unknown, misbehaving CRS participants, e.g. Proximax [56] distributes a subset of proxy addresses to a given user, and decides on further distribution based on whether or not previously distributed proxies are still available.

5.1.3 Lock-stepped Interactions (LI)

A censor can probe suspected proxies to confirm CRS participation. To mitigate this threat, a CRS can introduce a sequence of interactions to the client-proxy protocol that raises the cost of probing while hiding the true nature of the CRS proxy. This strategy is similar to rate limiting except now it involves censor-CRS interaction over the data channel whereas earlier only the dissemination channel was considered.

The main method is to introduce an authentication mechanism during data channel establishment between the client and overt destination. The properties of this technique are similar to those of proofs-of-work, except the CRS can monitor connections and at first feign non-CRS behavior to confound the censor's probes, e.g. seeing if a particular port is accepting connections. Only after a successful controlled interaction does the CRS expose itself, e.g. BridgeSPA [72], SkypeMorph [82], and DEFIANCE [61] all require clients to present authentication tokens. The censor needs to invest resources to conduct the interaction, and thus harvesting is less efficient, which again enables windows of opportunity during which the FNR can be maintained.

5.2 Detection Phase

In the detection phase, we assume the censor has failed to prevent users from learning the information needed to use the CRS initially. The censor's goal is to distinguish CRS traffic from other, allowed network flows. Censorship resistance strategies here focus on making the CRS data channel undetectable in practice, usually by avoiding low-cost, high-quality distinguishers, e.g. publicly known IP addresses of CRS participants. This takes advantage of the censor's resource limitations by raising the overall cost of processing network traffic and is especially effective in situations where real-time processing, i.e. at network line speed, is required to effectively detect and then block CRS traffic.

5.2.1 Obfuscated Values (OV)

This strategy aims to obfuscate distinguishers using techniques to hide their true values, often by using misdirection. On the data channel, common distinguishers are information in packet headers, such as IP addresses, and other traffic characteristics such as packet sizes.

By manipulating the data channel's path through the network, CRS systems can obfuscate the fact the client is communicating with a particular covert destination. Instead of directly connecting with the destination, the CRS client instead connects with an intermediary node (the overt destination) and from there is forwarded to the desired destination. A prime example of this is Tor [21], which bounces the data channel over three hops in a manner that ensures that no one—node, observer, or destination—can link the client with the destination. A recent alternative is *decoy routing* [40, 51, 84, 85], which typically works by including steganographic signals in packet headers that are part of a TLS connection between the client and an allowed, overt destination. Routers on the network path then process these signals and ensure the covert destination receives the client's request, either by diverting the client's flow from the overt destination or by simply forwarding a copy. The basic capabilities for decoy routing already exist in current networking equipment.

Effectively obfuscating CRS traffic patterns requires managing its network-level characteristics. For example, Scram-

bleSuit [82] removes the unique, 586-byte packet size feature from Tor traffic by padding to match generally occurring packet size distributions, but not any specific protocol or service. Obfsproxy [17] removes the distinguishing ciphersuite information in Tor's TLS handshake messages by encrypting the entire TLS session.

5.2.2 Indistinguishable Values (IV)

As an extension of the previous strategy, some systems attempt to avoid or remove problematic distinguishers; the CRS is successful if the censor must either allow or ham-fistedly disrupt both CRS and non-CRS traffic. That is, the censor may know that a non-CR system is being used for CRS activity, but be unable to distinguish CRS from non-CRS use cases. In practice, however, it is difficult, if not impossible, to achieve indistinguishability between the CRS and target non-CRS service data channels; we will defer an in-depth discussion of the limitations of IV strategies to subsection 5.4.

One approach is *mimicry*: the CRS identifies a non-CRS target distribution for a given high-quality distinguisher it wishes to remove, and then ensures that the values of the associated CRS feature conform to this target distribution. Here the onus is on the CRS design and implementation of *all* aspects of the data channel to conform to the expected non-CRS distributions, and the assumption is that this is both achievable and maintainable. An example is SkypeMorph [65], which shapes Tor traffic to look like that of a Skype video call. StegoTorus [79], format transforming encryption (FTE) [23], and Marionette [24] all transform Tor traffic to mimic other types of traffic, e.g. HTTP or Javascript. For higher-latency traffic, Collage [10], which extends the basic ideas of Infranet [28] and Message in a Bottle [47], uses steganographic techniques to embed CRS traffic in images posted to popular image hosting platforms.

Tunneling is another common technique, in which the CRS embeds the data channel in a non-CRS system and relies on this system for data transport. This method assumes that the introduction of CRS traffic does not cause a deviation in any non-CRS value distribution. Examples include meek [31], which leverages the fact that popular platforms such as Google's App Engine host multiple services and are all accessible from a common external address. However, within the encrypted traffic is the address of the true covert destination or location of content, which the platform knows about. This mechanism is known as *fronting* and is a common, legitimate technique in shared hosting scenarios. Similarly, CloudTransport [8] uses the storage platform S3 from Amazon as a read/write buffer in the cloud, allowing access to the CRS operating on a server elsewhere. Freewave [41] encodes data as audio and tunnels over Skype to hide the true nature of the traffic; Facet [60] also uses Skype but embeds YouTube videos into the video stream. Recent designs [38, 75] leverage online gaming platforms to make use of in-game chat features or game data manipulation as the CRS data channel.

5.3 Response Phase

Generally, the censor's *response* to the output of the detection phase is to block, interfere with, or allow suspicious traffic flows. The censor may also seek to punish CRS participants or to devote additional resources to analyzing the CRS in order to learn more effective distinguishers. To mitigate the effectiveness of the censor's response, CRS designs aim to ensure availability even in the case of successful detection and to provide plausible deniability for client and operators.

5.3.1 Churned Values (CV)

A CRS design can introduce *churn* by using values for only short lengths of time; the underlying idea is that a temporary feature-value pair is not effective as a distinguisher. The usual method is to choose a large pool of unpredictable values and then use and retire them within a short time frame. The censor needs to continuously employ resources at the exposure phase to determine the most recent CRS values. The advantage of churn over rate limiting is that in the latter, the censor's knowledge is cumulative, whereas in the former the censor's knowledge is quickly outdated.

Examples include FlashProxy, in which forwarders are temporary and last only for as long as a website client is viewing the website [30], and DEFIANCE [61], which cycles through a large pool of IPs that act as forwarders only briefly. The distinction between the two systems is that in the former, the value churn is determined by website visitors, which are extrinsic to the CRS, whereas in the latter the churn rate is managed by, and hence in complete control of, the CRS.

5.3.2 Outside the Sphere of Influence (OSoI)

CRSs can leverage inherent limitations of the censor's SoI by placing critical components beyond it. In this case, even if the censor can identify the CRS component as a target, the censor may well be unable to launch an effective attack. Almost all CRSs employ this strategy to some extent, usually by placing or using overt destinations and CRS information outside the SoI, to which clients are able to establish data and dissemination channels, respectively.

As an early example, Anderson's Eternity Service [1, 3] replicates content across a large number of overt destinations deployed in diverse jurisdictions, the idea being that at least some servers will be outside any given censor's scope of influence. An important variant of this strategy is to take advantage of existing, *CRS-agnostic*, third-party infrastructure or services instead of deploying their own, *CRS-specific*, infras-

tructure. Cloud providers like Google, Amazon, and CloudFlare, or VoIP providers like Skype, or content hosts like YouTube and Flickr, are all CRS-agnostic platforms that have been leveraged for their OSoI properties.

5.4 Limitations of the CRS Strategies

We now consider the limitations of the strategies presented above, providing details and discussion in proportion to the amount of research interest and literature that exists for particular strategies.

The major limitation of the almost-universally adopted strategy of putting critical infrastructure outside the censor's SoI (OSoI) is that the censor can simply block access to any infrastructure it can not shut down. Often, however, the collateral damage of blocking access would be minimal or acceptable to the censor. This limitation is the primary reason that the other strategies—ones that increase the collateral damage potential and target the data channel, such as OV and IV—have been developed to work in concert with OSoI and hence keep it relevant.

Manipulating distinguisher value distributions, a technique used for the UV, OV, and IV strategies, relies on characterizing the target distributions accurately and then carefully adhering to them, both difficult steps in practice. Often the cover traffic distribution is very different in character from that of the CRS traffic, and manipulation introduces inefficiencies, e.g. because padding is used, or there are conflicting constraints in the underlying network architecture, such as the CRS being unable to recruit IPs in the desired ranges.

Similarly, the CV strategy can be problematic, as 1) the CRS may not be in control of the relevant value space, and may be unable to manipulate the churn rate, such as in the case of FlashProxy, and 2) new values must constantly be available. If the churn rate is too high, then the client and CRS-management layer have an increased overhead of provisioning values for use, and also there is a danger of depleting the value pool.

Another limitation is that CRSs often require extensive management infrastructure, especially to deploy the RL, CV, and LI strategies. This raises the bar for deployment and usually requires increased communication and computation overhead for both the client and the CRS. For example, DEFIANCE [61] requires the client to maintain communications with the registration server for the duration of their usage of the system; any loss in connectivity with the server causes the data channel to fail.

A related issue, which mostly applies to the RL and LI strategies, is that the censor can sometimes deploy more resources than anticipated to negate these strategies. The Sybil attack is fairly common, for example, in which the censor deploys or controls a large enough pool of IP addresses that the benefits of rate limiting proxy IP address distribution are neutralized.

Specific to the decoy routing technique is that all of the implemented systems rely on the strategic deployment of the decoy router, with the assumption that this makes it impossible for the censor to "route around" cooperating ISPs without significant collateral damage. If this assumption is invalid, however, the censor can avoid or otherwise blackhole the route and nullify this CRS technique. Thomson et al. [69] investigate the feasibility of this attack while Houmansadr et al. [42] evaluate the costs associated with it. This is an open research problem, with game-theoretic analysis as an avenue for further pursuit.

Finally, the IV strategy, which uses mimicry and tunneling to protect the data channel, has seen a surge of activity recently, with cover protocols ranging from VoIP, to encrypted (TLS) web traffic, to streaming video. While in theory, from the censor's point of view, the CRS and legitimate CRS-agnostic service are indistinguishable with respect to the data channel, this is quite difficult to achieve in practice [34, 39, 60]. The main difficulty is that the censor only has to find one disparity, whereas a CRS must perfectly imitate the chosen cover protocol. Cover protocols are generally complex, with behavior dependent on particular use cases, and the CRS must not only make their imitation look correct, i.e. matches explicit values, but also ensure it behaves according to expected norms, i.e. matches implicit values. Common protocol-level disparities are a result of incomplete or incorrect cover protocol implementation, such as failure to handle errors in a consistent manner. Both SkypeMorph and CensorSpoofer, for example, suffer from systematic errors stemming from incomplete imitation of the Skype protocol [34, 39, 60].

Even if CRS traffic is tunneled over a cover protocol, the design may rely on channel characteristics in a different manner from that of the cover protocol.The censor may be able to take advantage of these inconsistencies in usage or content. Even if the data channel is encrypted, the content of CRS communications may still be distinguishable from that of the cover protocol, e.g. Li et al. [60] and Geddes et al. [34] detect Freewave using an n-gram based classifier on packet lengths. Moreover, the censor may not even need to identify CRS traffic: if the cover protocol is more robust to network degradation, for example, the censor can manipulate the network to disrupt CRS traffic while not affecting legitimate cover protocol traffic. Tor uses the TLS protocol to hide its signature, but expects its TLS connections to be longer-lived than the length of time required to load a typical text-based webpage; Iran used this fact to attack Tor by limiting the duration of TLS connections to two minutes [73]. Iran also throttled TLS connections to 2 kbps, making browsing and other activities over Tor difficult [58]. SWEET [86], which uses email as a carrier for web traffic, and Freewave [41], which uses Skype, are two more

examples of this vulnerability; email is a high-latency tolerant protocol and web traffic is not, and streaming audio/video is loss tolerant.

Such attacks, however, are at least contingent on the censor accurately modeling normal content. This may be easier for special-purpose protocols, such as VoIP, which is used only for video or voice traffic, than for general-purpose protocols, such as TLS, which is used for multiple types of data. In light of the difficulties of achieving consistency at the protocol, channel, and content levels, new approaches are necessary. Li et al. propose that CRSs not only use popular, unblocked cover protocols to tunnel traffic, but also to match CRS content with that of the chosen cover protocol. Their proposal, Facet, builds a data channel that sends video traffic over Skype and ensures the video traffic approximately matches the expected traffic pattern for a video chat call. It remains to be seen if further discrepancies may be discovered with this approach.

5.5 Properties of Existing CRS Designs

We now classify existing CRS designs with respect to the bold-faced security and performance properties defined in subsection 3.3. We limit the scope of our survey here by only including the subset of CRS designs that have clear descriptions in the literature, and that are not merely slight variations of an already-established design. In this way we can avoid misrepresenting the designs and goals of the CRSs considered as well as not miss out on significant classes of CRS designs. Table 2 presents the strategies used and the security and performance *profiles* (the set of properties that a CRS provides) provided by each of the CRSs we have considered and is a reference for the discussion that follows. We do not include a column for unblockable data channel, as every CRS design in the literature is by definition attempting to provide, albeit often unsuccessfully, an unblockable data channel through some means. Instead, we focus on those properties that help us distinguish CRS designs from each other.

A major drawback of current CRS design methodology is that most designs are stand-alone and implemented from scratch, usually with tightly integrated functionality; this precludes other systems from leveraging the effort already invested. Even designs that share a common base suffer from this shortcoming, e.g. the Tor network and the various pluggable transports for ensuring undetectable data channels. Although the Tor community is actively trying to address this problem [46] with the *pluggable transport (PT)* framework, the desired level of modularity and composabiity within PTs themselves are not currently in place [45, 52].

Only a few CRSs provide anonymity and coercion resistance for publishers, and these designs either debuted in the early 2000s or leverage the approach of the earlier systems. Research has shifted to providing read access to censored content, whereas the specter of chilled speech was at the forefront of earlier work. Recent systems, particularly those that rely on CRS-agnostic third-party platforms, often actually expose publishers, by requiring the publisher to interact directly with the third party, e.g. to set up accounts and/or store and manipulate files. This is not a consequence of some inherent limitation of CRS-agnostic designs, but rather that publisher anonymity is often not a design goal. Such designs can be improved by incorporating the anonymizing and coercion resistance techniques used in earlier systems, or by utilizing those systems directly.

Unlinkability is another common casualty of the recent trend of focusing on access at the cost of other security properties. A number of CRSs are intent only on bypassing blocking and employ direct communication between clients and overt/covert destinations. This situation can be ameliorated by using an appropriate path obfuscation design; certain CRS designs (e.g. Tor or Web MIXes) that provide publisher anonymity can be readily utilized for this purpose. However, these designs often have performance drawbacks, so the suitability of this approach depends on the use case.

With respect to performance, we observe a dichotomy among those systems promising high performance. While the potential throughput is comparable to that of unfettered Internet access, in reality only those designs that leverage CRS-agnostic platforms and services—which are provisioned for Internet-scale performance—are likely to actually enjoy this level of performance. The other designs rely on individual, often volunteer, nodes that are typically not capable of high throughput and are likely to achieve only middling levels of performance. We will revisit this topic in section 6, when we consider the implications of using third parties in CRS designs.

We see from Table 2 that no single CRS provides all of the security and performance properties we have identified—recall from subsection 5.4 some of the major limitations of the strategies employed by the designs in achieving these properties. However, if we consider hybrid designs, we observe that while it may be possible to increase coverage of desirable security properties by combining CRS systems, this is likely to come at a performance cost. For example, in the case of a CRS-agnostic design, if a client accesses the third party through a CRS that provides anonymity, such as Tor—which gives plausible deniability of CRS participation—then security property coverage can be improved, but performance will be poor due to the additional overhead of Tor's multi-hop routing mechanism. However, there can be cases where the hybrid's performance profile remains the same, such as a combination of an early publishing design, such as Publius, Eternity, or Tangler, with a system that use steganography to access content on popular platforms, such as Collage, Infranet, or MIAB. This kind of hybrid can be useful for use cases that must already allow for

Table 2. Overview of security and performance properties of various CRS designs and the strategies they use to achieve them. White squares denote that the system does not provide that security property or does not use that strategy, while black squares mean that it does. Dashes denote that the security property does not apply to that particular system. The systems, down the left, are grouped by the similarity of the strategies they mainly leverage. CRSs that utilize steganographic techniques are labeled with †, and those that leverage CRS-agnostic platforms are labeled with ∗.

UV	RL	LI	OV	IV	CV	OSoI	System		Year	Undetectable Data Channel	Anonymous Publisher	Unlinkable Client-Dest. Path	Incoercible Publisher	Incoercible Covert Dest.	Deniable Client	Deniable Forwarder	Deniable Covert Dest.	Reachability	Interactivity	Throughput	Goodput	Latency
■	□	□	□	□	□	■	TAZ-Rewebber	[36]	1998	□	□	■	■	■	□	□	□	□	■	■	■	■
■	□	□	□	□	□	■	Freenet	[12]	2000	□	■	■	■	■	□	□	□	□	□	□	□	□
■	□	□	□	□	□	■	Publius	[77]	2000	□	□	■	■	■	□	□	■	□	□	□	□	□
■	□	□	□	□	□	■	Tangler	[76]	2001	□	□	■	■	■	□	□	■	□	□	□	□	□
■	□	□	□	□	□	■	Serjantov	[71]	2002	□	■	■	■	■	□	□	□	■	□	□	□	□
■	□	■	■	□	□	■	Tor-Hidden Services	[21]	2004	□	■	■	■	■	□	□	□	□	■	■	■	■
■	■	□	□	□	□	■	Tor Bridges	[20]	2006	□	■	■	–	–	□	□	–	■	■	■	■	■
■	□	□	□	□	■	■	FlashProxy	[30]	2012	□	■	□	–	–	□	□	–	■	■	■	■	■
■	■	■	□	□	■	■	DEFIANCE	[61]	2012	□	■	□	–	–	□	□	–	■	■	■	■	■
□	■	□	□	□	□	■	Untrusted Messenger	[29]	2003	□	■	■	–	–	□	□	–	■	■	■	■	■
□	■	□	□	□	□	■	Köpsell and Hillig	[53]	2004	□	■	■	–	–	□	□	–	■	■	■	■	■
□	■	□	□	□	□	■	Unblock∗	[70]	2012	■	–	□	–	–	■	□	–	■	■	■	■	■
□	□	■	□	□	□	■	BridgeSPA	[72]	2011	–	–	–	–	–	□	■	–	–	–	–	–	–
□	□	■	■	□	□	■	ScrambleSuit	[82]	2013	■	–	–	–	–	□	□	–	■	■	■	□	■
□	□	□	■	□	□	■	TriangleBoy	[43]	2000	□	–	□	–	–	□	□	–	■	□	■	■	□
□	□	□	■	□	□	■	Web MIXes	[5]	2000	□	■	■	–	–	□	□	–	■	■	■	■	■
□	□	□	■	□	□	■	Tor	[21]	2004	□	■	■	–	–	□	□	–	■	■	■	■	■
□	□	□	■	□	□	■	Dust	[81]	2010	■	–	–	–	–	□	□	–	■	■	■	□	■
□	□	□	■	□	□	■	COR∗	[50]	2011	□	■	■	–	–	□	□	–	■	■	■	■	■
□	□	□	■	□	□	■	Telex	[85]	2011	■	–	□	–	–	■	□	–	□	■	■	■	■
□	□	□	■	□	□	■	Decoy Routing	[51]	2011	■	–	□	–	–	■	□	–	□	■	■	■	■
□	□	□	■	□	□	■	Cirripede	[40]	2011	■	–	□	–	–	■	□	–	□	■	■	■	■
□	□	□	■	□	□	■	Obfsproxy	[17]	2012	■	–	–	–	–	□	□	–	■	■	■	□	■
□	□	□	■	□	□	■	OSS∗	[32]	2013	■	–	■	–	–	■	■	–	■	■	□	□	□
□	□	□	■	□	□	■	TapDance	[84]	2014	■	–	□	–	–	■	□	–	□	■	■	■	■
□	□	□	■	□	□	□	DenaLi†	[67]	2014	■	–	■	–	–	■	■	–	□	■	□	□	■
□	□	□	□	■	□	■	Infranet†∗	[28]	2002	■	□	□	□	■	■	■	■	□	□	□	□	□
□	□	□	□	■	□	■	Collage†∗	[10]	2010	■	□	□	□	■	■	■	■	□	□	□	□	□
□	□	□	□	■	□	■	GoAgent∗	[35]	2011	■	–	□	□	■	■	■	□	■	■	■	■	■
□	□	□	□	■	□	■	StegoTorus†	[79]	2012	■	–	–	–	–	□	□	–	■	■	■	□	■
□	□	□	□	■	□	■	Freewave∗	[41]	2012	■	–	■	–	–	□	■	–	■	■	■	□	■
□	□	□	□	■	□	■	SkypeMorph	[65]	2012	■	–	■	–	–	□	□	–	■	■	■	□	■
□	□	□	□	■	□	■	CensorSpoofer	[78]	2012	■	–	□	–	–	□	□	–	■	□	■	■	□
□	□	□	□	■	□	■	MIAB†∗	[47]	2012	■	□	□	□	■	■	■	■	□	□	□	□	□
□	□	□	□	■	□	■	SWEET	[86]	2013	■	–	□	–	–	■	□	–	■	□	■	■	■
□	□	□	□	■	□	■	FTE	[23]	2013	■	–	–	–	–	□	□	–	■	■	■	□	■
□	□	□	□	■	□	■	Marionette	[24]	2015	■	–	–	–	–	□	□	–	■	■	■	□	■
□	□	□	□	■	□	■	CloudTransport∗	[8]	2013	■	–	□	–	–	■	■	–	■	■	■	■	■
□	□	□	□	■	□	■	Facet∗	[60]	2014	■	–	□	–	–	□	□	–	□	□	■	□	■
□	□	□	□	■	□	■	Facade†	[49]	2014	■	□	□	□	■	■	■	■	□	□	■	□	□
□	□	□	□	■	□	■	meek∗	[31]	2015	■	–	□	□	■	■	■	□	■	■	■	■	■
□	□	□	□	■	□	■	Castle†∗	[38]	2015	■	□	□	□	■	■	■	■	□	■	□	□	■
□	□	□	□	■	□	■	Rook†∗	[75]	2015	■	□	□	□	■	■	■	■	□	■	□	□	■
□	□	□	□	□	□	■	Eternity	[1]	1996	□	□	□	■	■	□	□	□	□	□	■	■	■
Strategies										Security Properties								Performance				

performance compromises, but now more security properties are present.

The preceding discussion reveals that CRS designs are typically forced to make a trade-off between security and performance, and while most attempt to secure design components and mitigate attack vectors, the strategies employed suffer from important limitations, as discussed in subsection 5.4. Additionally, our analysis reveals certain gaps in the security provided by the spectrum of strategies in the literature. The efforts that do exist to combat these weaknesses are mostly on a case-by-case basis; there is no systematic approach of which we are aware. First, there are no effective protections against the censor poisoning CRS information, such as routing information [69]. Any public information used by both CRS and non-CRS activities, however, has at least an implicit level of defense, since it is not without risk for the censor to poison such information. Second, most CRSs do not yet prevent crushing tactics, such as a DDoS, against CRS-

participating destinations, storage, or network resources [18]. The CRS implicitly depends on the leveraged platform or their network connectivity provider to prevent such a scenario; in general, DDoS attacks do not yet have a robust solution, outside of over-provisioning of bandwidth and IP addresses. Third, corrupted, malicious content and CRS participants are not adequately defended against and is an area of active research [6, 34].

At present, the primary work to address these problems is in the Tor ecosystem. The poisoning of the public relay information, such as IP address and port numbers, is mitigated using digital signature schemes, but protecting the authenticity and confidentiality (from the censor) of bridge addresses is still an outstanding problem. There have been many denial-of-service attacks, both theoretical and actual, on the Tor network; these have been addressed on a case-by-case basis through programmatic changes. Finally, the Tor network actively attempts to detect suspicious relays through various network-level tests, but this is again done in an ad-hoc fashion.

6 Looking Forward

Recent work focuses on raising the bar for the censor to detect the data channel by mitigating perceived low-cost distinguishers. However, the capabilities and willingness of a given real-world censor to engage in sophisticated traffic analysis is unclear, and the cost of detecting complex distinguishers is not well understood. While the literature does analyze detection attacks, these analyses are usually implementation specific, and as such the results are of limited scope and difficult to relate to a realistic censor. There is a need for Internet-scale studies of distinguisher effectiveness, which may provide clues about which distinguishers are of the highest utility to the censor and the biggest threats to the defender.

The trend of forgoing publisher security in favor of performance assumes that targeted attacks and surveillance are the major threat, and that it is not realistic to believe that a majority of clients will be targeted or harmed. However, recent revelations, in particular those by Edward Snowden [68], call these design choices into question. The dramatic reach of "Five Eyes"—a program of cooperation and surveillance data sharing between the governments of Australia, Canada, New Zealand, the United Kingdom, and the United States—necessitates a reevaluation of the basic assumptions most CRSs make with respect to the censor's sphere of influence and visibility. Systems are typically not designed to withstand global passive adversaries, for instance, and designs often count on distribution across diverse jurisdictions to make this assumption realistic. Given that a significant number of government entities cooperate and have far-reaching network capabilities, systems designed with these assumptions may be more brittle than previously believed. While it is unclear just how large the spheres of visibility and influence are for any given censor, their reach is likely far larger than anticipated by current CRS designs.

Another potentially problematic trend is the increasing reliance on existing, popular third-party infrastructure and data channels, based on a belief in the effectiveness of collateral damage. Indeed, we note that the indistinguishable values strategy is becoming more and more prevalent, and data-channel protection techniques are seemingly moving from CRS-specific to CRS-agnostic designs. Whereas CRS-specific implementations are crafted to meet all design specifications and updated as necessary, CRS-agnostic implementations meet those same specifications through happy, and possibly temporary, coincidence. As a result, there are extrinsic properties we must account for when evaluating such a CRS. Even without the censor's interference, this may be undesirable. First, this type of design has dependencies on external parties who may not be invested in the CRS's success. Second, the CRS is relying on properties that do not exist by its own design, which means there may be unexpected states that render the CRS ineffective, or the needed functionality may simply disappear with an update. Finally, the fate of the CRS is tied to the third party, and how widely the service is adopted, potentially limiting the client base of the CRS.

We now revisit the notion of collateral damage, which plays an essential role in design of data channels and selection of overt destinations for CRSs. The theory goes: if the CRS picks the perfect third-party service or platform to leverage, one that the censor balks at blocking due to the inherent collateral damage that is contingent on the limitations of its classifier, then CRS activity will go unmolested. If the CRS is able to mostly blend in, then the censor will be unable to cleanly remove CRS activity, and will be forced to tolerate some CRS activity. In this case, the CRS causes the CRS-agnostic services and platforms to act as *concentration* points for maximizing collateral damage potential.

Counterintuitively, the censor may actually benefit from this concentration, since the attack surfaces and CRS security failures are also concentrated and well defined, e.g. CloudTransport uses only traffic to Amazon's cloud storage and hence is easier to contain. We present three cases in which CRS-agnostic service or platform concentration induces negative CRS outcomes.

First, the CRS-agnostic service or platform is now a single point of failure, which means that if the censor does decide to block it, perhaps because the CRS designer overestimated the potential collateral damage, or despite it, the CRS is effectively contained, while the impact is limited to the CRS-agnostic service or platform only.

Second, the censor or entities within its SoI may develop local alternatives for the targeted CRS-agnostic service and draw legitimate CRS-agnostic users away, leaving CRS users exposed, e.g. Weibo, YouKu, and Baidu.[2] Such services are more easily controlled by the censor and unlikely to be successfully leveraged by a CRS.

Finally, CRS-agnostic services and platforms that have been leveraged are often operated by single corporations. The censor can strike back at CRS systems by attacking the CRS-agnostic service or platform and/or the entity that operates it. The strike can be in the form of actual network-level attacks [62] that cause the operator to reconsider or decry [22] its role as a host to CRS activity. Indeed, there is uncertainty around exactly how the operating entity may respond. In the worst case it may even act as an informant to the censor and monitor CRS activity. This is most troubling when we recall that many high-performing CRS-agnostic designs do not provide anonymity or unlinkability. Since the long-term viability of leveraging third parties by CRSs is unclear, it is critical that these security properties be in place.

The picture this paints is that concentration is a poor direction, since it has synergy with the censor's desire for containment. We should reexamine the trend of leveraging CRS-agnostic entities and platforms as a shield, as this causes the aforementioned extrinsic factors to become entangled with the intrinsic CRS design. As we see from the discussion above, if collateral damage is to remain a deterrent then it must not come from the concentration provided by an easily contained entity.

As a mitigation, we propose *diffusion* as a remedy: CRS designs ought to leverage ubiquitous Internet protocols, such as TLS and network topologies that avoid single points of failure, e.g. peer-to-peer. Diffusion avoids providing a constrained set of stakeholders for the censor to target and pressure, as ubiquitous technologies belong to no one and everyone. As a result, the censor should have a difficult time containing CRS activity to some portion or subset of the network value space, and be unable to effectively mitigate the impact of collateral damage.

As future work we need to investigate the implicit assumption that collateral damage is an overwhelming factor in the censor's decision function. Current CRS schemes are contingent on and try to maximize collateral damage (FPR), but do not evaluate the impact of information leakage (FNR) on censor behavior, nor identify parameters that might affect it. Filling this gap is an important next step in illuminating the dynamics of the censorship resistance game and providing feedback on best practices for CRS design. Indeed, we are beginning to see activity in this vein in recent work by Tchantz et al. [74] and Elahi et al. [26]. The latter specifically pursues the last two areas of research, by applying game-theoretic analysis to censorship resistance and investigating the impact of information leakage, collateral damage, and accuracy of the censorship apparatus on the censor's behavior.

7 Related Work

There are previous attempts to systematize the censorship resistance field, but these are more limited in scope than our work.

Köpsell and Hillig [53] present a *classification of blocking criteria* from the point of view of the censor and its decision-making process, and also provide general *blocking circumvention strategies*. While their main aim is to present a censorship circumvention solution in the network traffic blocking setting, their classification sheds light into the motivations and decision-making process of the censor and is valuable for this reason.

They classify blocking (or censorship) decision-making criteria according to the TCP/IP protocol layer of the communication and whether if it is based on the *content* or *circumstances* of said communication. They classify the *circumstances* of communication into addresses, timing, data transfer, and services. Examples of these include: source and destination addresses, send and receive times and connection duration, bandwidth and latency, and the protocols used. The *content*, where deep packet inspection is employed, is categorized by kind, properties, type and value. The kind of communication can be an object or a stream; its properties can include whether it is encrypted, compressed or other statistics. The type can tell us if it is an image, audio, or a binary, and the values are the byte patterns in the transmission that can be detected. It is feasible that as technology progresses, deep packet inspection will provide further identifiers that the censor could leverage.

They consider blocking circumvention as concerned with two challenges. On the one hand is the *infrastructure* of the blocking circumvention service and on the other is the *distribution* of the information about the blocking infrastructure. For the former they identify the "many access points" and the "all or nothing" circumvention strategies. They give the example of deploying a large number of access points which the censor must counter completely or else have failed in censoring effectively. They identify the "out-of-band" strategy observing that the communication requirements for distributing information *about* the service are far lower than for traffic flowing through it. They also identify techniques for limiting

2 We do not claim that these alternatives are a direct effort to quash CRS activity, but as the local user base of the CRS-agnostic service thins, potential collateral damage also decreases.

the censor's ability to harvest information about the "many access points" but limit the discussion to their key-space hopping proposal.

We extend their network traffic blocking criteria to encompass all the attack surfaces offered by the Internet and circumvention systems and provide a more thorough look at the strategies employed by a wider range of blocking circumvention systems and the fundamental basis of those strategies.

The most closely related work to ours is a draft, from 2010, where Leberknight et al. [55] provide a survey of censorship and circumvention from technological and political dimensions. They provide a taxonomy of circumvention technologies, and discuss the design features that are critical for success. While similar in theme, our works diverge in several points. We treat the technological aspects of censorship and censorship circumvention in greater depth, while they focus in equal parts on the social and political as well. They only consider anonymity, content protection and content filtering as the challenges to overcome while we show that there are more attack surfaces for the censor to target. The techniques they discuss are limited to their identified challenges and the taxonomy they provide is similarly restricted to those aspects of censorship circumvention. We provide a thorough treatment of the attack surfaces and censor capabilities, and provide more breadth and depth in terms of the censorship circumvention techniques and the fundamental basis for them. Finally, since 2010, many novel systems have been developed that have added to the censorship circumvention toolkit and thus our work provides a more contemporary account of the state of the art.

The most recent related work is by Khattak et al. [52]. They focus exclusively on the pluggable transports framework and how it may be extended to allow for more modularity and reuse. They propose "tweakable transport" as a generalized framework that can help alleviate the incompatibilities that exist today. Their model of pluggable transports mirrors our own in that they are also concerned with the threat of detection of the data channel. However, we differ in scope since we also consider the censorship apparatus, CRS use-cases, and a broader range of attack surfaces.

These works provide useful insights into the censorship circumvention space, both from the censor and circumvention system perspectives. We leverage these classifications as appropriate and augment them to produce a more robust and all-inclusive classification of censorship circumvention strategies and techniques.

8 Conclusion

We have presented a systematization of knowledge of Internet censorship resistance, which provides a simple yet useful model of the censor apparatus and its inherent imperfect nature and the common strategies and techniques found in the CRs literature. Our model illuminates how the strategies apply upward pressure on the censor's error rates.

Our discussion also identifies likely areas for future work. We noted that reuse and modularity were lacking and identified Tor pluggable transports as one effort to address this shortcoming for the Tor ecosystem. The current state is not at the desired level but there is progress in the right direction. An active research avenue focuses on the need for empirical data about real-world censorship with respect to CRS user impact and the censorship techniques used [9]. What few investigations exist are informative, but provide only partial and possibly outdated snapshots of the state of censorship. In particular, many CRS designs assume the censor is capable and willing to engage in sophisticated traffic analysis, as we noted in section 6. As discussed earlier, a thorough analysis to evaluate censor attacks on distinguishers is needed. We also lack research into the relationship between the many CRS techniques and the costs to the censor, particularly in terms of policy and decision making. While there is some recent activity [26, 74] in this area, further attention should be paid to game-theoretic analysis of optimal strategies, which would be helpful in understanding the censor decision function and identify useful techniques for CRS designs.

Our investigation uncovers certain attack vectors, such as network DDoS, participant corruption, and information poisoning that are as yet not addressed in a holistic and strategic manner.

We noted the rising trend of leveraging popular CRS-agnostic services and platforms as a means of mitigating censor actions through collateral damage in the response phase, and called out three potential problems. First, leveraging CRS-agnostic services tend to concentrate the attack surface and that this helps the censor better contain the CRS. Second, the extent of the censor's SoI and SoV is perhaps underestimated, which we have reason to believe due to recent revelations, and utilizing CRS-agnostic services also provides the means for mass surveillance, which we note is another means to promote censorship. Finally, these CRS-agnostic services rely on properties extrinsic to the CRS itself. This reliance makes it difficult to evaluate CRS designs that leverage them—the CRS cannot control them and hence one cannot know if the security and performance profiles will remain consistent and the CRS functional. We proposed *diffusion*, which is the return to distribution of risk over many entities, stakeholders, and attack surfaces.

The systematization of knowledge presented in this paper enables a more principled approach to designing and evaluating censorship resistance systems. Our aim is that the results of this paper will lead to better and more robust designs that

address the gaps we have identified and help the censorship resistance field in effectively managing and getting ahead in its ever-escalating arms race.

References

[1] R. Anderson. The Eternity Service. In *Proceedings of Pragocrypt*, 1996.

[2] S. Aryan, H. Aryan, and J. A. Halderman. Internet Censorship in Iran: A First Look. In *Proceedings of the USENIX Workshop on Free and Open Communications on the Internet*. USENIX, 2013.

[3] A. Back. Usenet Eternity. *Phrack Magazine*, http://www.cypherspace.org/eternity/phrack.html, 1997. Retrieved May 2015.

[4] R. Barnes, A. Cooper, and O. Kolkman. Technical Considerations for Internet Service Filtering. IETF-Draft, http://tools.ietf.org/html/draft-iab-filtering-considerations-01, 2012.

[5] O. Berthold, H. Federrath, and S. Köpsell. Web MIXes: A System for Anonymous and Unobservable Internet Access. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 115–129. Springer-Verlag, LNCS 2009, July 2000.

[6] A. Biryukov, I. Pustogarov, and R.-P. Weinmann. Trawling for Tor Hidden Services: Detection, Measurement, Deanonymization. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, May 2013.

[7] R. M. Bond, C. J. Fariss, J. J. Jones, A. D. Kramer, C. Marlow, J. E. Settle, and J. H. Fowler. A 61-Million-Person Experiment in Social Influence and Political Mobilization. *Nature*, 489(7415):295–298, 2012.

[8] C. Brubaker, A. Houmansadr, and V. Shmatikov. CloudTransport: Using Cloud Storage for Censorship-Resistant Networking. In *Proceedings of 14th Privacy Enhancing Technologies Symposium*. Springer, 2014.

[9] S. Burnett and N. Feamster. Making Sense of Internet Censorship: A New Frontier for Internet Measurement. *ACM SIGCOMM Computer Communication Review*, 43(3):84–89, July 2013.

[10] S. Burnett, N. Feamster, and S. Vempala. Chipping Away at Censorship Firewalls with User-Generated Content. In *Proceedings of the 19th USENIX Security Symposium*, 2010.

[11] D. L. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[12] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. In *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 46–66, July 2000.

[13] R. Clayton, S. J. Murdoch, and R. N. M. Watson. Ignoring the Great Firewall of China. In G. Danezis and P. Golle, editors, *Proceedings of the Sixth Workshop on Privacy Enhancing Technologies*, pages 20–35. Springer, June 2006.

[14] A. Dainotti, C. Squarcella, E. Aben, K. C. Claffy, M. Chiesa, M. Russo, and A. Pescapé. Analysis of Country-wide Internet Outages Caused by Censorship. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, pages 1–18. ACM, 2011.

[15] R. Dingledine. Iran blocks Tor; Tor Releases Same-Day Fix. *Tor Blog*, https://blog.torproject.org/blog/iran-blocks-tor-tor-releases-same-day-fix, September 2011. Retrieved May 2015.

[16] R. Dingledine. Research Problems: Ten Ways to Discover Tor Bridges. *Tor Blog*, https://blog.torproject.org/blog/research-problems-ten-ways-discover-tor-bridges, October 2011. Retrieved May 2015.

[17] R. Dingledine. Obfsproxy: The Next Step in the Censorship Arms Race. *Tor Blog*, https://blog.torproject.org/blog/obfsproxy-next-step-censorship-arms-race, February 2012. Retrieved May 2015.

[18] R. Dingledine. How to Handle Millions of New Tor Clients. https://blog.torproject.org/blog/how-to-handle-millions-new-tor-clients, September 2013. Retrieved August 2015.

[19] R. Dingledine, N. Hopper, G. Kadianakis, and N. Mathewson. One Fast Guard for Life (or 9 Months). In *7th Workshop on Hot Topics in Privacy Enhancing Technologies*, 2014.

[20] R. Dingledine and N. Mathewson. Design of a Blocking-Resistant Anonymity System. Technical Report 2006-1, The Tor Project, November 2006.

[21] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[22] E. Dou and A. Barr. U.S. Cloud Providers Face Backlash From China's Censors. *The Wall Street Journal*, http://www.wsj.com/articles/u-s-cloud-providers-face-backlash-from-chinas-censors-1426541126, 2015. Retrieved May 2015.

[23] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton. Protocol Misidentification Made Easy with Format-Transforming Encryption. In *Proceedings of the 20th ACM conference on Computer and Communications Security*, November 2013.

[24] K. P. Dyer, S. E. Coull, and T. Shrimpton. Marionette: A Programmable Network Traffic Obfuscation System. In *Proceedings of the 24th USENIX Security Symposium*, pages 367–382. USENIX Association, August 2015.

[25] T. Elahi, K. Bauer, M. AlSabah, R. Dingledine, and I. Goldberg. Changing of the Guards: A Framework for Understanding and Improving Entry Guard Selection in Tor. In *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*, pages 43–54. ACM, 2012.

[26] T. Elahi, J. Doucette, H. Hosseini, S. Murdoch, and I. Goldberg. A Framework for the Game-theoretic Analysis of Censorship Resistance. Technical Report 2015-11, CACR, 2015. http://cacr.uwaterloo.ca/techreports/2015/cacr2015-11.pdf.

[27] N. Eltantawy and J. Wiest. The Arab Spring | Social Media in the Egyptian Revolution: Reconsidering Resource Mobilization Theory. *International Journal of Communication*, 5(0), 2011.

[28] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. Karger. Infranet: Circumventing Web Censorship and Surveillance. In *Proceedings of the 11th USENIX Security Symposium*, August 2002.

[29] N. Feamster, M. Balazinska, W. Wang, H. Balakrishnan, and D. Karger. Thwarting Web Censorship with Untrusted

Messenger Delivery. In R. Dingledine, editor, *Proceedings of Privacy Enhancing Technologies workshop*, pages 125–140. Springer-Verlag, LNCS 2760, March 2003.

[30] D. Fifield, N. Hardison, J. Ellithorpe, E. Stark, R. Dingledine, P. Porras, and D. Boneh. Evading Censorship with Browser-Based Proxies. In *Proceedings of the 12th Privacy Enhancing Technologies Symposium*. Springer, July 2012.

[31] D. Fifield, C. Lan, R. Hynes, P. Wegmann, and V. Paxson. Blocking-resistant Communication through Domain Fronting. *Proceedings on Privacy Enhancing Technologies*, 2015(2):46–64, June 2015.

[32] D. Fifield, G. Nakibly, and D. Boneh. OSS: Using Online Scanning Services for Censorship Circumvention. In *Proceedings of the 13th Privacy Enhancing Technologies Symposium*, July 2013.

[33] K. Fisher. The Death of SuprNova.org. *Ars Technica*, http://arstechnica.com/staff/2005/12/2153/, December 2005. Retreived Nov 2012.

[34] J. Geddes, M. Schuchard, and N. Hopper. Cover Your ACKs: Pitfalls of Covert Channel Censorship Circumvention. In *Proceedings of the 20th ACM conference on Computer and Communications Security*, 2013.

[35] goagent. GoAgent. *Pseudonymously*, https://github.com/goagent/goagent, July 2011. Retrieved May 2015.

[36] I. Goldberg and D. Wagner. TAZ Servers and the Rewebber Network: Enabling Anonymous Publishing on the World Wide Web. *First Monday*, 3(4), August 1998.

[37] D. Goodin. Massive denial-of-service attack on GitHub tied to Chinese government. *Ars Technica*, http://arstechnica.com/security/2015/03/massive-denial-of-service-attack-on-github-tied-to-chinese-government/, March 2015. Retrieved May 2015.

[38] B. Hahn, R. Nithyanand, P. Gill, and R. Johnson. Games Without Frontiers: Investigating Video Games as a Covert Channel. http://arxiv.org/pdf/1503.05904v2.pdf, 2015. Retrieved May 2015.

[39] A. Houmansadr, C. Brubaker, and V. Shmatikov. The Parrot is Dead: Observing Unobservable Network Communication. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, May 2013.

[40] A. Houmansadr, G. T. K. Nguyen, M. Caesar, and N. Borisov. Cirripede: Circumvention Infrastructure using Router Redirection with Plausible Deniability. In *Proceedings of the 18th ACM conference on Computer and Communications Security*, October 2011.

[41] A. Houmansadr, T. Riedl, N. Borisov, and A. Singer. IP over Voice-over-IP for Censorship Circumvention. *arXiv preprint arXiv:1207.2683*, 2012.

[42] A. Houmansadr, E. L. Wong, and V. Shmatikov. No Direction Home: The True Cost of Routing Around Decoys. In *Network and Distributed System Security*. The Internet Society, 2014.

[43] S. Hsu. TriangleBoy. http://www.webrant.com/safeweb_site/html/www/tboy_whitepaper.html, 2000. Retrieved May 2015.

[44] ICANN Security and Stability Advisory Committee. Impacts of Content Blocking via the Domain Name System. http://www.icann.org/en/groups/ssac/documents/sac-056-en.pdf, October 2012. ICANN SSAC security advisory.

[45] ifinity0. Complete Specification for Generalised PT Composition. *Pseudonymously*, https://trac.torproject.org/projects/tor/ticket/10061, October 2013. Retrieved May 2015.

[46] ifinity0. Composing Pluggable Transports. *Pseudonymously*, https://github.com/infinity0/tor-notes/blob/master/pt-compose.rst, October 2013. Retrieved May 2015.

[47] L. Invernizzi, C. Kruegel, and G. Vigna. Message in a Bottle: Sailing Past Censorship. In *Privacy Enhancing Technologies Symposium*, 2012.

[48] A. Johnson, C. Wacek, R. Jansen, M. Sherr, and P. Syverson. Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries. In *Proceedings of the 20th ACM conference on Computer and Communications Security*, November 2013.

[49] B. Jones, S. Burnett, N. Feamster, S. Donovan, S. Grover, S. Gunasekaran, and K. Habak. Facade: High-Throughput, Deniable Censorship Circumvention Using Web Search. In *Proceedings of the USENIX Workshop on Free and Open Communications on the Internet*. USENIX, 2014.

[50] N. Jones, M. Arye, J. Cesareo, and M. J. Freedman. Hiding Amongst the Clouds: A Proposal for Cloud-based Onion Routing. In *Proceedings of the USENIX Workshop on Free and Open Communications on the Internet*, August 2011.

[51] J. Karlin, D. Ellard, A. W. Jackson, C. E. Jones, G. Lauer, D. P. Mankins, and W. T. Strayer. Decoy Routing: Toward Unblockable Internet Communication. In *Proceedings of the USENIX Workshop on Free and Open Communications on the Internet*, August 2011.

[52] S. Khattak, L. Simon, and S. J. Murdoch. Systemization of Pluggable Transports for Censorship Resistance. http://arxiv.org/pdf/1412.7448v1.pdf, 2014. Retrieved May 2015.

[53] S. Köpsell and U. Hillig. How to Achieve Blocking Resistance for Existing Systems Enabling Anonymous Web Surfing. In *Proceedings of the Workshop on Privacy in the Electronic Society*, October 2004.

[54] C. Labovitz. Egypt Returns to the Internet. https://asert.arbornetworks.com/egypt-returns-to-the-internet/, February 2011. Retrieved August 2015.

[55] C. S. Leberknight, M. Chiang, H. V. Poor, and F. Wong. A Taxonomy of Internet Censorship and Anti-Censorship. http://www.princeton.edu/~chiangm/anticensorship.pdf, December 2010. Draft.

[56] K. Levchenko and D. McCoy. Proximax: Fighting Censorship With an Adaptive System for Distribution of Open Proxies. In *Proceedings of Financial Cryptography and Data Security*, February 2011.

[57] A. Lewman. Tor Partially Blocked in China. *Tor Blog*, https://blog.torproject.org/blog/tor-partially-blocked-china, September 2009. Retrieved May 2015.

[58] A. Lewman. New Blocking Activity from Iran. *Tor Blog*, https://blog.torproject.org/blog/new-blocking-activity-iran, January 2011. Retrieved May 2015.

[59] A. Lewman. Iran Partially Blocks Encrypted Network Traffic. *Tor Blog*, https://blog.torproject.org/blog/iran-partially-blocks-encrypted-network-traffic, February 2012. Retrieved May 2015.

[60] S. Li, M. Schliep, and N. Hopper. Facet: Streaming over Videoconferencing for Censorship Circumvention. In *Proceedings of the Workshop on Privacy in the Electronic Society*, November 2014.

[61] P. Lincoln, I. Mason, P. Porras, V. Yegneswaran, Z. Weinberg, J. Massar, W. A. Simpson, P. Vixie, and D. Boneh. Bootstrapping Communications into an Anti-Censorship System. In *Proceedings of the USENIX Workshop on Free and Open*

Communications on the Internet, August 2012.

[62] B. Marczak, N. Weaver, J. Dalek, R. Ensafi, D. Fifield, S. McKune, A. Rey, J. Scott-Railton, R. Deibert, and V. Paxson. China's Great Cannon. https://citizenlab.org/2015/04/chinas-great-cannon/, 2015. Retrieved May 2015.

[63] martin. China listening in on Skype - Microsoft assumes you approve. *Pseudonymously*, https://en.greatfire.org/blog/2012/dec/china-listening-skype-microsoft-assumes-you-approve, December 2012. Retrieved May 2015.

[64] D. McPherson. When Hijacking the Internet. https://asert.arbornetworks.com/when-hijacking-the-internet/, November 2008. Retrieved May 2015.

[65] H. Mohajeri Moghaddam, B. Li, M. Derakhshani, and I. Goldberg. SkypeMorph: Protocol Obfuscation for Tor Bridges. In *Proceedings of the 19th ACM conference on Computer and Communications Security*, October 2012.

[66] Z. Nabi. The Anatomy of Web Censorship in Pakistan. In *Proceedings of the USENIX Workshop on Free and Open Communications on the Internet*. USENIX, 2013.

[67] A. Narain, N. Feamster, and A. C. Snoeren. Deniable Liaisons. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 525–536. ACM, 2014.

[68] Paul Farrell. History of 5-Eyes – Explainer. *The Guardian*, http://www.theguardian.com/world/2013/dec/02/history-of-5-eyes-explainer, December 2013. Retrieved May 2015.

[69] M. Schuchard, J. Geddes, C. Thompson, and N. Hopper. Routing Around Decoys. In *Computer and Communications Security*. ACM, 2012.

[70] W. Scott, R. Cheng, J. Li, A. Krishnamurthy, and T. Anderson. Blocking-Resistant Network Services using Unblock. http://unblock.cs.washington.edu/unblock.pdf, October 2012. Retrieved May 2015.

[71] A. Serjantov. Anonymizing Censorship Resistant Systems. In *Proceedings of the 1st International Peer To Peer Systems Workshop*, March 2002.

[72] R. Smits, D. Jain, S. Pidcock, I. Goldberg, and U. Hengartner. BridgeSPA: Improving Tor Bridges with Single Packet Authorization. In *Proceedings of the Workshop on Privacy in the Electronic Society*. ACM, October 2011.

[73] Y. Torbati. Iranians Face New Internet Curbs Before Presidential Election. *Reuters*, http://www.reuters.com/article/2013/05/21/net-us-iran-election-internet-idUSBRE94K0ID20130521, May 2013. Retrieved May 2015.

[74] M. C. Tschantz, S. Afroz, V. Paxson, and J. Tygar. On Modeling the Costs of Censorship. *arXiv preprint arXiv:1409.3211*, 2014.

[75] P. Vines and T. Kohno. Rook: Using Video Games as a Low-Bandwidth Censorship Resistant Communication Platform. http://homes.cs.washington.edu/~yoshi/papers/tech-report-rook.pdf, 2015. Retrieved May 2015.

[76] M. Waldman and D. Mazières. Tangler: A Censorship-Resistant Publishing System based on Document Entanglements. In *Proceedings of the 8th ACM Conference on Computer and Communications Security*, pages 126–135, November 2001.

[77] M. Waldman, A. Rubin, and L. Cranor. Publius: A Robust, Tamper-Evident, Censorship-Resistant and Source-Anonymous Web Publishing System. In *Proceedings of the 9th USENIX Security Symposium*, pages 59–72, August 2000.

[78] Q. Wang, X. Gong, G. T. K. Nguyen, A. Houmansadr, and N. Borisov. CensorSpoofer: Asymmetric Communication using IP Spoofing for Censorship-Resistant Web Browsing. In *Proceedings of the 19th ACM conference on Computer and Communications Security*, October 2012.

[79] Z. Weinberg, J. Wang, V. Yegneswaran, L. Briesemeister, S. Cheung, F. Wang, and D. Boneh. StegoTorus: A Camouflage Proxy for the Tor Anonymity System. In *Proceedings of the 19th ACM conference on Computer and Communications Security*, October 2012.

[80] T. Wilde. Knock Knock Knockin' on Bridges' Doors. *Tor Blog*, https://blog.torproject.org/blog/knock-knock-knockin-bridges-doors, January 2012. Retrieved June 2015.

[81] B. Wiley. Dust: A Blocking-Resistant Internet Transport Protocol. http://blanu.net/Dust.pdf, 2011. Retrieved May 2015.

[82] P. Winter, T. Pulls, and J. Fuss. ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship. In *Proceedings of the Workshop on Privacy in the Electronic Society*. ACM, November 2013.

[83] S. Wolchok, R. Yao, and J. A. Halderman. Analysis of the Green Dam censorware system. *Computer Science and Engineering Division, University of Michigan*, 18, 2009.

[84] E. Wustrow, C. M. Swanson, and J. A. Halderman. TapDance: End-to-Middle Anticensorship Without Flow Blocking. In *Proceedings of the 23rd USENIX conference on Security Symposium*, pages 159–174. USENIX Association, 2014.

[85] E. Wustrow, S. Wolchok, I. Goldberg, and J. A. Halderman. Telex: Anticensorship in the Network Infrastructure. In *Proceedings of the 20th USENIX Security Symposium*, August 2011.

[86] W. Zhoun, A. Houmansadr, M. Caesar, and N. Borisov. SWEET: Serving the Web by Exploiting Email Tunnels. *HotPETS*, 2013.

[87] T. Zhu, D. Phipps, A. Pridgen, J. R. Crandall, and D. S. Wallach. The Velocity of Censorship: High-Fidelity Detection of Microblog Post Deletions. In *Proceedings of the 22nd USENIX Security Symposium*. USENIX, 2013.