Threat modeling and circumvention of Internet censorship

David Fifield

September 27, 2017

Abstract

Research on Internet censorship is hampered by a lack of adequate models of censor behavior, encompassing both censors' current practice and their likely future evolution. Censor models guide the development of circumvention systems, so it is important to get them right. A censor model should be understood not only as a set of capabilities—such as the ability to monitor network traffic—but also as a set of priorities constrained by resource limitations. A circumvention system designed under inadequate assumptions runs the risk of being either easily blocked, or impractical to deploy.

My thesis research will be concerned with developing empirically informed censor models and practical, effective circumvention systems to counter them. My goal is to move the field away from seeing the censorship problem as a cat-and-mouse game that affords only incremental and temporary advancements. We should instead state the hypotheses and assumptions under which our circumvention designs will work—with the designs being more or less practical depending on how well the hypotheses and assumptions match the behavior of real-world censors.

1 Thesis

My research is about Internet censorship and how to make it ineffective. To this end, I am interested in building useful models of real-world censors as they exist today and may exist in the future, for the purpose of building circumvention systems that are not only sound in theory but also effective in practice.

2 Scope

Internet censorship is an enormous topic. My thesis research is concerned with one important case of it: the border firewall. A user resides within a censor's "sphere of influence," within which the censor controls all network links. The user's goal is to *circumvent* the censor's controls; that is, to access some blocked destination outside the censor's sphere of influence. Circumvention requires somehow safely traversing a hostile network in order to reach a destination. The censor can block direct access to any destination, so circumvention typically uses, at minimum, some kind of indirect access, such as connecting through a proxy server. This scope is deliberately underspecified, in order to leave room for variations; filling in the details is the heart of threat modeling.

The scope is motivated by important real-world use cases, most notably the control of the network by the government of a country, with the sphere of influence being the country's borders. "Country" is a useful heuristic for a censor's sphere of influence: a government typically has the power to enforce laws and control network infrastructure to act within its own borders, but not outside. However, even under the auspices of a national government, a "censor" may not be a uniform body. Restrictions may vary across geographic locations, and enforcement may be distributed across many ISPs who are more or less compliant. A censor's power may not stop strictly at its borders—international business deals, for example, may enlarge the scope of entities who may be assumed to collude with the censor. This research is also concerned with censorship at a smaller scale, for instance commercial firewall boxes controlling access out of a local network. The reason for considering these is that they differ from national firewalls in interesting ways (with regard to whitelisting versus blacklisting, for example), and the capabilities of commercial networking technology may presage those of larger censors in the future. In fact, the study of censorship and circumvention is closely related to that of network monitoring in general.

The scope excludes other important cases of censorship, for example that which occurs entirely within the censor's sphere of influence, or anything not involving the Internet. So, for example, we do not consider the blocking of keywords on China's domestic microblogging service Weibo, and censorship of television and printed newspapers. We also leave out the important and difficult topic of self-censorship—people choosing not to express themselves, in whatever medium, for fear of repercussion. We do, however, consider the possibility of non-network-based attacks by censors, such as physical violence or imprisonment, but only as it relates to circumvention of a border firewall.

3 Studies of censors

The main tool we have to build relevant threat models is the natural study of censors. The study of censors is complicated by difficulty of access: censors are not forthcoming about their methods. Researchers are obligated to treat censors as a black box, drawing inferences about their internal workings from their externally visible characteristics. The easiest thing to learn is the censor's *what*—the destinations that are blocked. Somewhat harder is the investigation into *where* and *how*, the specific technical mechanisms used to effect censorship and where they are deployed in the network. What we are really interested in, and what is most difficult to infer, is the *why*, or the motivations and goals that underlie a censorship apparatus. We posit that censors, far from being unitary entities of focused purpose, are rather complex organizations of people and machines, with conflicting purposes and economic rationales, subject to resource limitations. The *why* gets to the heart of why circumvention is even possible: a censoring firewall's duty is not merely to block, but to *discriminate* between what is blocked and what is allowed, in support of some other goal. Circumvention systems confuse this discrimination in order to sneak traffic through the firewall.

Past measurement studies have mostly been short-lived, one-off affairs, focusing deeply on one region of the world for at most a few months. Thus published knowledge about censors' capabilities consists mostly of a series of "spot checks" with blank areas between them. There have been a few designs proposed to do ongoing measurements of censorship, such as ConceptDoppler [10] in 2007 and CensMon [38] in 2011, but these have not lasted long in practice, and for the most part there is an unfortunate lack of longitudinal and cross-country measurements. Just as in circumvention, in censorship measurement a host of difficulties arise when running a scalable system for a long time, that do not arise when doing a one-time operation. The situation is thankfully becoming better, with the increasing data collection capabilities of measurement systems like OONI [21].

A long description of past measurement studies appears in Appendix A, based on a survey done by me and others in 2016 [40, §IV.A].

From the survey of measurement studies we may draw some general conclusions. Censors change over time, sometimes for unknown reasons, and not always in the direction of greater restrictions. Censorship conditions differ greatly across countries, not only in subject but in mechanism and motivation. The "Great Firewall" of China has long been the world's most sophisticated censor, but it is in many ways an outlier, and not representative of censors elsewhere. Most censors are capable of manipulating DNS

responses, IP address blocking, and keyword filtering at some level.

A reasonable set of capabilities, therefore, that a contemporary censor may be assumed to have is:

- blocking of specific IP addresses and ports,
- control of default DNS servers,
- injection of false DNS responses,
- injection of TCP RSTs,
- throttling of connection,
- keyword filtering
- protocol identification, and
- temporary total shutdown of Internet connections

Not all censors will be able to do all of these. As the amount of traffic to be handled increases, in-path attacks such as throttling become relatively more expensive. Whether a particular censoring act even makes sense will depend on a local cost–benefit analysis. Some censors may be able to tolerate a brief total shutdown, while for others the importance of the Internet is too great for such a crude measure.

Past measurement studies have done a good job at determining the technical aspects of censorship, for example where in the network censorship routers are located. There is not so much known about the inner workings of censors. The anonymous paper on China's DNS censorship [4] probably comes closest to the kind of insight I am talking about, with its clever use of side channels to infer operational characteristics of censor boxes. For example, their research found that each DNS injection node runs a few hundred independent processes. This is indirect information, to be sure, but it hints at the level of resources the censor is able to bring to bear. I am interested in even deeper information, for example how censors make the decision on what to block, and what bureaucratic and other considerations might cause them to work less than optimally.

4 Principles of circumvention

The purpose of threat modeling is, in my view, to enable the building of more effective circumvention. We study censors in order to learn how to defeat them.

Seen from the censor's point of view, censorship is a classification problem. There is some class of traffic that the censor wants to block, but also there is traffic that the censor prefers not to block—whatever authority controls the censor must see *some* benefit to allowing Internet access. For each packet, stream (or whatever), the censor must make a decision about whether to block or allow. Circumvention can be understood as making this classification problem more difficult, of increasing the cost of misclassifications.

When the censor fails to block something that it would have preferred to block, it is a misclassification called a *false negative*. The cost of false negative classifications is an increase in whatever a censor wishes to suppress through censorship: for example popular demonstrations, free journalism, and political organizing. When the censor blocks something that it would have preferred to allow, the misclassification is a *false positive*. The cost of false positive misclassifications is a diminishing in the utility of the Internet: people and businesses just trying to get on with their work or lives encounter obstacles, generally decreasing productivity and other qualities the censor might want to preserve.

The cost of false positives is so important to circumvention that researchers have a specialized term for it: collateral damage. Collateral damage encompasses all the harm suffered by the censor through inadvertent, ancillary blocking done in the course of censorship. The term is a bit unfortunate, because it is easily misunderstood. If circumventors do things right, the potential "damage" is never realized, because the censor sees the cost as being too great. Circumventors try to make false positives so expensive that the censor has no choice but to allow false negatives; that is, to permit circumvention traffic.

Early censors (around the time of the late 1990s and early 2000s) would be considered weak by today's standards. They were mostly easy to circumvent by simple countermeasures, such as tweaking a protocol or using an alternative DNS server. But as censors become more capable, our models have to evolve to match. Indeed, my interest in threat modeling might be described as a sort of meta-modeling, learning about how threat models change over time and according to circumstances.

It is helpful to categorize the challenges of circumvention into three parts. The first is blocking by content; that is, by what you say. HTTP request keyword filtering and blocking based on deep packet inspection fall into this category. The second is blocking by address; that is, by whom you talk to. IP address blocking and DNS tampering fall into this category. The third is active probing, in which the censor imitates a client in order

to discover proxy servers. Active probing is usually used as input for an address-blocking mechanism. Of these challenges, address blocking is probably the hardest, because it is efficient to implement in firewall hardware, and because network addresses are a scarcer resource than protocol variations.

Appendix B contains a summary of censorship circumvention systems and how they have changed over time in response to changing censorship threats.

5 My research

There are three themes to my research under this proposal:

1. Building circumvention systems according to evolving censor models.
2. Empirical testing of real-world censors to generate better models.
3. Evaluating existing circumvention systems against realistic models.

Each theme is covered in a subsection below.

5.1 Building circumvention systems

Over the past five years I have been involved in the development of four noteworthy circumvention designs:

- Flash proxy [17], based on temporary proxies running in web browsers.
- OSS [19], using third-party web scanning services.
- Domain fronting [18], using popular web services for cover.
- Snowflake [39, 16] (in progress), based on peer-to-peer proxies in web browsers; flash proxy redux.

These have evolved according to the needs of the time and my growing understanding of how censorship should be modeled.

My main interest is resistance to address blocking, which I regard as more difficult to achieve than resistance to content blocking. The first two systems, flash proxy and OSS, made no special effort to avoid their content being blocked, leaving content obfuscation to be done by another layer. My later designs have taken the threats of content blocking and active probing more integrally into account.

5.1.1 Flash proxy, a circumvention system

I began working on censorship circumvention with flash proxy in 2011. Flash proxy is targeted at the difficult problem of proxy address blocking: it is designed against a censor model in which the censor can block any IP address it chooses, but only on a relatively slow timeline of several hours.

Flash proxy works by running tiny JavaScript proxies in ordinary users' web browsers. The mini-proxies serve as temporary stepping stones to a full-fledged proxy, such as a Tor relay. The idea is that the flash proxies are too numerous, diverse, and quickly changing to block effectively. A censored user may use a particular proxy for only seconds or minutes before switching to another. If the censor manages to block the IP address of one proxy, there is little harm, because many other temporary proxies are ready to take its place.

The flash proxy system was designed under interesting constraints imposed by being partly implemented in JavaScript in the browser. The proxies sent and received data using the WebSocket protocol, which allows for socket-like persistent TCP connections in browsers, but with a catch: the browser can only make outgoing connections, not receive incoming ones as a traditional proxy would. The censored client must somehow inform the system of its own public address, and then the proxy connects *back* to the client. This architectural constraint was probably the biggest impediment to the usability of flash proxy, because it required users to configure their local router to permit incoming connections. (Removing this impediment is the main reason for the development of Snowflake, described later.) Flash proxy does not itself try to obfuscate patterns in the underlying traffic; it only provides address diversity.

For the initial "rendezvous" step in which a client advertises its address and a request for proxy service, flash proxy uses a neat idea: a low-capacity, but highly covert channel bootstraps the high-capacity, general-purpose WebSocket channel. For example, we implemented an automated email-based rendezvous, in which the client would send its address in an encrypted email to a special address. While it is difficult to build a useful low-latency bidirectional channel on top of email, email is difficult to block and it is only needed once, at the beginning of a session. We later replaced the email-based rendezvous with one based on domain fronting, which would later inspire meek, described below.

I was the leader of the flash proxy project and the main developer of its code. Flash proxy was among the first circumvention systems built for Tor—only obfs2 is older. It was first deployed in Tor Browser in January

2013, and was later retired in January 2016 after it ceased to see appreciable use. Its spirit lives on in Snowflake, now under development.

Flash proxy appeared in the 2012 research paper "Evading Censorship with Browser-Based Proxies" [17], which I coauthored with Nate Hardison, Jonathan Ellithorpe, Emily Stark, Roger Dingledine, Phil Porras, and Dan Boneh.

5.1.2 OSS, a circumvention prototype

OSS, for "online scanning service," is a design for circumvention based on the use of third-party web services that issue HTTP requests to user-specified destinations, such as an online translation service. OSS is designed against the model of a censor that is unwilling to block useful web services that are used for circumvention, because of the useful service they provide.

In OSS, the client sends messages to a censored destination by bouncing them through a third-party scanning service. The key idea is a deliberate conflation of address and content. The client asks the scanning service to scan a long URL that is constructed to encode both the destination host and a data payload. The destination receives the HTTP request and decodes its payload. The destination sends data downstream by abusing HTTP redirection, instructing the scanning service to send another HTTP request back to the client, with a different payload. The resistance to blocking of the OSS system hinges on the abundance of online scanning services that exist.

OSS was never deployed to users. I judged its overhead and potential to annoy webmasters to be too great to be practical. The core idea, however, did see use as a rendezvous method for flash proxy. In this method, a helper program would encode the client's IP address into a URL. The user would then copy and paste the URL into any online scanning service, which would then forward the information to the flash proxy system. In fact, this URL encoding was used internally by the domain fronting–based rendezvous as well, using a URL as a convenient vehicle for data transfer.

OSS appeared in the 2013 research paper "OSS: Using Online Scanning Services for Censorship Circumvention" [19], which I coauthored with Gabi Nakibly and Dan Boneh.

5.1.3 Domain fronting and meek

My most influential contribution to the world of circumvention is my research on domain fronting. While the basic idea is not mine, the research

I led and the code I wrote helped domain fronting become the ubiquitous tool it is today.

Domain fronting assumes a rather strong censor model, essentially equivalent to the state of the art of national censors at the time of its popularization. That is, a censor that can block IP addresses and domain names, that can filter plaintext HTTP, can fingerprint protocol implementations. The main censor capabilities not provided for are probabilistic classification by traffic flow characteristics, and high-collateral-damage blocking of HTTPS on important web servers. What I find most intellectually compelling about domain fronting research is that is finally begins to transcend the "cat-and-mouse" paradigm that has plagued thinking around circumvention, and to put blocking resistance on a scientific basis. By this I mean that one can state assumptions, and consequences that hold as long as the assumptions are true. For example, we do not make claims such as "domain fronting is unblockable"; rather, we may state hypotheses and consequents: "if fronting through a domain with sufficient collateral damage, such that the censor is unwilling to block it, and if the censor does not find some side channel that distinguishes fronted from non-fronted traffic, then the communication will be unblocked." This kind of thinking, that of weighing censors' *costs* and *capabilities*, underlies my thinking about threat modeling.

Like flash proxy, domain fronting is primarily targeted at the problem of address blocking (though it is effective against content blocking and active probing as well). The core idea is the use of different domain names at different layers of communication. The "outside" layers, those visible to the censor, contain an innocuous "front" domain name, ideally one that is hard to block because of the value of the services behind it. The "inside" layer, invisible to the censor under encryption, contains the true, presumably censored, destination. An intermediate server, whose name is the front domain name, removes the outer layer of encryption and forwards the information to the covert destination. There are a number of important services that support domain fronting, mainly cloud providers and content delivery networks. On top of this basic machinery, it is relatively easy to build a general-purpose covert bidirectional communications channel, one that can even be made reasonably efficient.

I wrote and continue to maintain the code of meek, a circumvention transport for Tor based on domain fronting. It first appeared in Tor Browser in October 2014, and continues operation to the present. My code has been forked and incorporated by other circumvention projects, notably including Psiphon and Lantern, with whom I continue to collaborate. Today, meek is Tor's second-most-used transport, carrying around 10 terabytes of user

traffic each month.

Domain fronting appeared in the 2015 research paper "Blocking-resistant communication through domain fronting" [18], which I coauthored with Chang Lan, Rod Hynes, Percy Wegmann, and Vern Paxson.

5.1.4 Snowflake, a circumvention system

I am working on a new circumvention system, a transport for Tor called Snowflake. Snowflake is the successor to flash proxy. It keeps the basic idea of in-browser proxies while fixing the usability problems that hampered the adoption of flash proxy. My main collaborators in this project are Serene Han and Arlo Breault.

The key difference between flash proxy and Snowflake is the basic communications protocol between client and browser proxy. Flash proxy used the TCP-based WebSocket protocol, which required users to configure their personal firewall to allow incoming connections. Snowflake instead uses WebRTC, a UDP-based protocol that enables peer-to-peer connections without manual configuration. The most similar existing system is uProxy [41], which in one of its operating modes uses WebRTC to connect through a friend's computer. Snowflake differs because it does not require prior coordination with a friend before connecting. Instead, it pulls its proxies from a pool of web users who are running the Snowflake code. Beyond the changed protocol, we hope to build in performance and efficiency improvements.

Snowflake will afford interesting research opportunities. One, of course, is the design of the system itself—no circumvention system of its nature has previously been deployed at a large scale. Another opportunity is observing how censors react to a new challenge.

Most of the available documentation on Snowflake is linked from the project's wiki page [39]. Mia Gil Epner and I wrote a preprint on the fingerprinting hazards of WebRTC [16].

5.2 Empirically testing real-world censors

In 2015 I helped study the phenomenon of "active probing" by the Great Firewall to discover hidden proxy servers. In active probing, the censor pretends to be a legitimate client of the proxy server: it connects to suspected servers to check whether they speak a proxy protocol. If they do, then they are blocked. Active probing makes good sense for the censor: it has high precision (low risk of collateral damage), and is efficient because it can be run as a batch job apart from a firewall's real-time responsibilities.

The Great Firewall can dynamically active-probe and block the servers of a number of common circumvention protocols, such as Tor, obfs2, and obfs3, within only seconds or minutes of a connection by a legitimate client. The need to resist active probing has informed the design of recent circumvention systems, including meek.

My primary contribution to the active probing project was the analysis of server logs to uncover the history of about two and a half years of active probing. My work revealed the wide distribution of active probing source addresses (there were over 14,000 of them). It also discovered previously undocumented types of probes, for the protocol used by VPN Gate and for a simple form of domain-fronted proxy. I helped analyze the network "fingerprints" of active probes and how they might be distinguished from connections by legitimate clients.

The work on active probing appeared in the 2015 research paper "Examining How the Great Firewall Discovers Hidden Circumvention Servers" [14], which I coauthored with Roya Ensafi, Philipp Winter, Nick Feamster, Nicholas Weaver, Vern Paxson.

I am interested in understanding censors at a deeper level. To that end, I am working on a project to measure how long censors take to react to sudden changes in circumvention. So far, our technique has been to monitor the reachability of newly added Tor Browser bridges, to see how long after they are introduced they get blocked. Portions of this work have already appeared in the 2016 research paper "Censors' Delay in Blocking Circumvention Proxies" [20], which I coauthored with Lynn Tsai. We discovered some interesting, previously undocumented behaviors of the Great Firewall of China. While the firewall, through active probing, is able to detect some bridges dynamically within seconds or minutes, it lags in detecting Tor Browser's newly added bridges, taking days or weeks to block them. It seems that bridges are first blocked only at certain times of day, perhaps reflecting an automated batch operation.

I am now continuing to work on this project along with Lynn Tsai and Qi Zhong. We plan to run targeted experiments to find out more about how censors extract bridge addresses from public information, for example, by adding bridges with different attributes and seeing whether they are blocked differently. Our first experiment used measurement sites only in China and Iran, but we hope to expand to many more countries by collaborating with measurement platforms such as OONI [21] and ICLab [24]. We hope to solicit other kinds of censor delays from other circumvention projects, in order to build a more all-encompassing picture of censors' priorities with respect to circumvention.

5.3 Evaluating circumvention against realistic censor models

Evaluating the quality of circumvention systems is tricky, whether they are only proposed or actually deployed. The problem of evaluation is directly tied to threat modeling. Circumvention is judged according to how well it works under a given model; the evaluation is therefore meaningful only as far as the threat model reflects reality. Without grounding in reality, researchers risk running an imaginary arms race that evolves independently of the real one.

This kind of work is rather different than the direct evaluations of circumvention tools that have happened before, for example those done by the Berkman Center [37] and Freedom House [6] in 2011. Rather than testing tools against censors, we evaluated how closely calibrated designers' own models were to models derived from actual observations of censors.

This research was partly born out of frustration with some typical assumptions made in academic research on circumvention, which we felt placed undue emphasis on steganography and obfuscation of traffic streams, while not paying enough attention to the perhaps more important problems of bridge distribution and rendezvous. Indeed, in our survey of over 50 circumvention tools, we found that academic designs tended to be concerned with detection in the steady state after a connection is established, while actually deployed systems cared more about how the connection is established initially. We wanted to help bridge the gap by laying out a research agenda to align the incentives of researchers with those of circumventors. This work was built on extensive surveys of circumvention tools, measurement studies, and known censorship events against Tor.

This work on evaluation appeared in the 2016 research paper "Towards Grounding Censorship Circumvention in Empiricism" [40], which I coauthored with Michael Carl Tschantz, Sadia Afroz, and Vern Paxson.

6 Out-of-scope research

For completeness I list here projects that have to do with censorship, but do not fit exactly into the scope of this thesis proposal.

"Do you see what I see? Differential treatment of anonymous users" [29], written with Sheharbano Khattak, Sadia Afroz, Mobin Javed, Srikanth Sundaresan, Vern Paxson, Steven J. Murdoch, and Damon McCoy. In this, we studied *server-side* censorship, discrimination by a server against a client, in the absence of a censor in the middle.

"Tor's Usability for Censorship Circumvention" [31], written with Linda

Lee, Nathan Malkin, Ganesh Iyer, Serge Egelman, and David Wagner. We conducted a pair of user studies to see how well people were able to configure Tor Browser to connect in a simulated censorship environment, then recommended changes to the user interface.

7 Projected schedule

Fall 2016

- qualifying exam
- project on censor delay
- development of Snowflake

Spring 2017

- dissertation writing
- deployment of Snowflake
- courses, 10-hour GSI

Summer 2017

- dissertation writing
- tracking Snowflake deployment

Fall 2017

- dissertation writing
- courses
- graduation

References

[1] Collin Anderson. *Dimming the Internet: Detecting Throttling as a Mechanism of Censorship in Iran*. Tech. rep. University of Pennsylvania, 2013. URL: http://arxiv.org/pdf/1306.4361v1.pdf.

[2] Collin Anderson, Philipp Winter, and Roya. "Global Network Interference Detection over the RIPE Atlas Network". In: *Free and Open Communications on the Internet*. USENIX, 2014. URL: https://www.usenix.org/system/files/conference/foci14/foci14-anderson.pdf.

[3] Yawning Angel and Philipp Winter. *obfs4 (The obfourscator)*. May 2014. URL: https://gitweb.torproject.org/pluggable-transports/obfs4.git/tree/doc/obfs4-spec.txt.

[4] Anonymous. "Towards a Comprehensive Picture of the Great Firewall's DNS Censorship". In: *Free and Open Communications on the Internet*. USENIX, 2014. URL: https://www.usenix.org/system/files/conference/foci14/foci14-anonymous.pdf.

[5] Simurgh Aryan, Homa Aryan, and J. Alex Halderman. "Internet Censorship in Iran: A First Look". In: *Free and Open Communications on the Internet*. USENIX, 2013. URL: http://censorbib.nymity.ch/pdf/Aryan2013a.pdf.

[6] Cormac Callanan, Hein Dries-Ziekenheiner, Alberto Escudero-Pascual, and Robert Guerra. *Leaping Over the Firewall: A Review of Censorship Circumvention Tools*. Tech. rep. Freedom House, 2011. URL: https://freedomhouse.org/report/special-reports/leaping-over-firewall-review-censorship-circumvention-tools.

[7] Abdelberi Chaabane, Terence Chen, Mathieu Cunche, Emiliano De Cristofaro, Arik Friedman, and Mohamed Ali Kaafar. "Censorship in the Wild: Analyzing Internet Filtering in Syria". In: *Internet Measurement Conference*. ACM, 2014. URL: http://conferences2.sigcomm.org/imc/2014/papers/p285.pdf.

[8] Richard Clayton. "Failures in a Hybrid Content Blocking System". In: *Privacy Enhancing Technologies*. Springer, 2006, pp. 78–92. URL: http://www.cl.cam.ac.uk/~rnc1/cleanfeed.pdf.

[9] Richard Clayton, Steven J. Murdoch, and Robert N. M. Watson. "Ignoring the Great Firewall of China". In: *Privacy Enhancing Technologies*. Springer, 2006, pp. 20–35. URL: http://www.cl.cam.ac.uk/~rnc1/ignoring.pdf.

[10] Jedidiah R. Crandall, Daniel Zinn, Michael Byrd, Earl Barr, and Rich East. "ConceptDoppler: A Weather Tracker for Internet Censorship". In: *Computer and Communications Security*. ACM, 2007, pp. 352–365. URL: http://www.csd.uoc.gr/~hy558/papers/conceptdoppler.pdf.

[11] Alberto Dainotti, Claudio Squarcella, Emile Aben, Kimberly C. Claffy, Marco Chiesa, Michele Russo, and Antonio Pescapé. "Analysis of Country-wide Internet Outages Caused by Censorship". In: *Internet Measurement Conference*. ACM, 2011, pp. 1–18. URL: http://conferences.sigcomm.org/imc/2011/docs/p1.pdf.

[12] Maximillian Dornseif. "Government mandated blocking of foreign Web content". In: *DFN-Arbeitstagung über Kommunikationsnetze*. Gesellschaft für Informatik, 2003, pp. 617–647. URL: https://subs.emis.de/LNI/Proceedings/Proceedings44/GI-Proceedings.44.innen-1.pdf.

[13] Kevin P. Dyer, Scott E. Coull, Thomas Ristenpart, and Thomas Shrimpton. "Protocol Misidentification Made Easy with Format-Transforming Encryption". In: *Computer and Communications Security*. ACM, 2013. URL: http://eprint.iacr.org/2012/494.pdf.

[14] Roya Ensafi, David Fifield, Philipp Winter, Nick Feamster, Nicholas Weaver, and Vern Paxson. "Examining How the Great Firewall Discovers Hidden Circumvention Servers". In: *Internet Measurement Conference*. ACM, 2015. URL: http://conferences2.sigcomm.org/imc/2015/papers/p445.pdf.

[15] Roya Ensafi, Philipp Winter, Abdullah Mueen, and Jedidiah R. Crandall. "Analyzing the Great Firewall of China Over Space and Time". In: *Privacy Enhancing Technologies* 2015.1 (2015). URL: http://censorbib.nymity.ch/pdf/Ensafi2015a.pdf.

[16] David Fifield and Mia Gil Epner. "Fingerprintability of WebRTC". In: *CoRR* abs/1605.08805 (2016). URL: https://arxiv.org/abs/1605.08805.

[17] David Fifield, Nate Hardison, Jonathan Ellithorpe, Emily Stark, Roger Dingledine, Phil Porras, and Dan Boneh. "Evading Censorship with Browser-Based Proxies". In: *Privacy Enhancing Technologies Symposium*. Springer, 2012, pp. 239–258. URL: http://crypto.stanford.edu/flashproxy/flashproxy.pdf.

[18] David Fifield, Chang Lan, Rod Hynes, Percy Wegmann, and Vern Paxson. "Blocking-resistant communication through domain fronting". In: *Privacy Enhancing Technologies* 2015.2 (2015). URL: http://www.icir.org/vern/papers/meek-PETS-2015.pdf.

[19] David Fifield, Gabi Nakibly, and Dan Boneh. "OSS: Using Online Scanning Services for Censorship Circumvention". In: *Privacy Enhancing Technologies Symposium*. Springer, 2013. URL: http://freehaven.net/anonbib/papers/pets2013/paper_29.pdf.

[20] David Fifield and Lynn Tsai. "Censors' Delay in Blocking Circumvention Proxies". In: *Free and Open Communications on the Internet*. USENIX, 2016. URL: https://www.usenix.org/system/files/conference/foci16/foci16-paper-fifield.pdf.

[21] Arturo Filastò and Jacob Appelbaum. "OONI: Open Observatory of Network Interference". In: *Free and Open Communications on the Internet*. USENIX, 2012. URL: https://www.usenix.org/system/files/conference/foci12/foci12-final12.pdf.

[22] Amir Houmansadr, Giang T. K. Nguyen, Matthew Caesar, and Nikita Borisov. "Cirripede: Circumvention Infrastructure using Router Redirection with Plausible Deniability". In: *Computer and Communications Security*. ACM, 2011, pp. 187–200. URL: http://hatswitch.org/~nikita/papers/cirripede-ccs11.pdf.

[23] Amir Houmansadr, Thomas Riedl, Nikita Borisov, and Andrew Singer. "I want my voice to be heard: IP over Voice-over-IP for unobservable censorship circumvention". In: *Network and Distributed System Security*. The Internet Society, 2013. URL: http://people.cs.umass.edu/~amir/papers/FreeWave.pdf.

[24] *ICLab*. URL: https://iclab.org/.

[25] George Kadianakis and Nick Mathewson. *obfs2 (The Twobfuscator)*. Jan. 2011. URL: https://gitweb.torproject.org/pluggable-transports/obfsproxy.git/tree/doc/obfs2/obfs2-protocol-spec.txt.

[26] George Kadianakis and Nick Mathewson. *obfs3 (The Threebfuscator)*. Jan. 2013. URL: https://gitweb.torproject.org/pluggable-transports/obfsproxy.git/tree/doc/obfs3/obfs3-protocol-spec.txt.

[27] Josh Karlin, Daniel Ellard, Alden W. Jackson, Christine E. Jones, Greg Lauer, David P. Mankins, and W. Timothy Strayer. "Decoy Routing: Toward Unblockable Internet Communication". In: *Free and Open Communications on the Internet*. USENIX, 2011. URL: http://static.usenix.org/event/foci11/tech/final_files/Karlin.pdf.

[28] Sheharbano Khattak, Tariq Elahi, Laurent Simon, Colleen M. Swanson, Steven J. Murdoch, and Ian Goldberg. "SoK: Making Sense of Censorship Resistance Systems". In: *Privacy Enhancing Technologies* 2016.4 (2016), pp. 37–61. URL: http://www.degruyter.com/downloadpdf/j/popets.2016.2016.issue-4/popets-2016-0028/popets-2016-0028.xml.

[29] Sheharbano Khattak, David Fifield, Sadia Afroz, Mobin Javed, Srikanth Sundaresan, Vern Paxson, Steven J. Murdoch, and Damon McCoy. "Do you see what I see? Differential treatment of anonymous users". In: *Network and Distributed System Security*. The Internet Society, 2016. URL: https://www.cl.cam.ac.uk/~sk766/publications/ndss16_tor_differential.pdf.

[30] Sheharbano Khattak, Mobin Javed, Philip D. Anderson, and Vern Paxson. "Towards Illuminating a Censorship Monitor's Model to Facilitate Evasion". In: *Free and Open Communications on the Internet*. USENIX, 2013. URL: http://censorbib.nymity.ch/pdf/Khattak2013a.pdf.

[31] Linda Lee, David Fifield, Nathan Malkin, Ganesh Iyer, Serge Egelman, and David Wagner. "Tor's Usability for Censorship Circumvention". MA thesis. EECS Department, University of California, Berkeley, May 2016. URL: https://www.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-58.html.

[32] Graham Lowe, Patrick Winters, and Michael L. Marcus. *The Great DNS Wall of China*. Tech. rep. New York University, 2007. URL: https://cs.nyu.edu/~pcw216/work/nds/final.pdf.

[33] Bill Marczak, Nicholas Weaver, Jakub Dalek, Roya Ensafi, David Fifield, Sarah McKune, Arn Rey, John Scott-Railton, Ron Deibert, and Vern Paxson. "An Analysis of China's "Great Cannon"". In: *Free and Open Communications on the Internet*. USENIX, 2015. URL: https://www.usenix.org/system/files/conference/foci15/foci15-paper-marczak.pdf.

[34] Zubair Nabi. "The Anatomy of Web Censorship in Pakistan". In: *Free and Open Communications on the Internet*. USENIX, 2013. URL: http://censorbib.nymity.ch/pdf/Nabi2013a.pdf.

[35] OpenNet Initiative. *Internet Filtering in China in 2004-2005: A Country Study*. URL: https://opennet.net/studies/china.

[36] Jong Chun Park and Jedidiah R. Crandall. "Empirical Study of a National-Scale Distributed Intrusion Detection System: Backbone-Level Filtering of HTML Responses in China". In: *Distributed Computing Systems*. IEEE, 2010, pp. 315–326. URL: http://www.cs.unm.edu/~crandall/icdcs2010.pdf.

[37] Hal Roberts, Ethan Zuckerman, and John Palfrey. *2011 Circumvention Tool Evaluation*. Tech. rep. Berkman Center for Internet and Society, Aug. 2011. URL: https://cyber.law.harvard.edu/publications/2011/2011_Circumvention_Tool_Evaluation.

[38] Andreas Sfakianakis, Elias Athanasopoulos, and Sotiris Ioannidis. "CensMon: A Web Censorship Monitor". In: *Free and Open Communications on the Internet*. USENIX, 2011. URL: http://static.usenix.org/event/foci11/tech/final_files/Sfakianakis.pdf.

[39] *Snowflake*. URL: https://trac.torproject.org/projects/tor/wiki/doc/Snowflake.

[40] Michael Carl Tschantz, Sadia Afroz, Anonymous, and Vern Paxson. "SoK: Towards Grounding Censorship Circumvention in Empiricism". In: *Symposium on Security & Privacy*. IEEE, 2016. URL: https://www.eecs.berkeley.edu/~sa499/papers/oakland2016.pdf.

[41] *uProxy*. URL: https://www.uproxy.org/.

[42] John-Paul Verkamp and Minaxi Gupta. "Inferring Mechanics of Web Censorship Around the World". In: *Free and Open Communications on the Internet*. USENIX, 2012. URL: https://www.usenix.org/system/files/conference/foci12/foci12-final1.pdf.

[43] Zachary Weinberg, Jeffrey Wang, Vinod Yegneswaran, Linda Briesemeister, Steven Cheung, Frank Wang, and Dan Boneh. "StegoTorus: A Camouflage Proxy for the Tor Anonymity System". In: *Computer and Communications Security*. ACM, 2012. URL: http://www.frankwang.org/papers/ccs2012.pdf.

[44] Philipp Winter and Stefan Lindskog. "How the Great Firewall of China is Blocking Tor". In: *Free and Open Communications on the Internet*. USENIX, 2012. URL: https://www.usenix.org/system/files/conference/foci12/foci12-final2.pdf.

[45] Sebastian Wolfgarten. "Investigating large-scale Internet content filtering". MA thesis. Dublin City University, Aug. 2006. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.133.5778&rep=rep1&type=pdf.

[46] Joss Wright. *Regional Variation in Chinese Internet Filtering*. Tech. rep. University of Oxford, 2012. URL: http://papers.ssrn.com/sol3/Delivery.cfm/SSRN_ID2265775_code1448244.pdf?abstractid=2265775&mirid=3.

[47] Eric Wustrow, Scott Wolchok, Ian Goldberg, and J. Alex Halderman. "Telex: Anticensorship in the Network Infrastructure". In: *USENIX Security Symposium*. USENIX, 2011. URL: http://www.usenix.org/event/sec11/tech/full_papers/Wustrow.pdf.

[48] Xueyang Xu, Z. Morley Mao, and J. Alex Halderman. "Internet Censorship in China: Where Does the Filtering Occur?" In: *Passive and Active Measurement Conference*. Springer, 2011, pp. 133–142. URL: http://web.eecs.umich.edu/~zmao/Papers/china-censorship-pam11.pdf.

A Past measurement studies of censors

This section surveys past measurement studies in order to draw specific and general conclusions about censor models. The objects of this survey are based on those in the evaluation study done by me and others in 2016 [40, §IV.A].

One of the earliest technical studies of censorship occurred not in some illiberal place, but in the German state of North Rhein-Westphalia. In 2003, Dornseif [12] tested ISPs' implementation of a controversial legal order to block two Nazi web sites. While there were many possible ways to implement the block, none were trivial to implement, nor free of overblocking side effects. The most popular implementation used *DNS tampering*, simply returning (or injecting) false responses to DNS requests for the domain names of the blocked sites. An in-depth survey of DNS tampering found a variety of implementations, some blocking more and some blocking less than required by the order.

Clayton [8] in 2006 studied a "hybrid" blocking system, called "CleanFeed" by the British ISP BT, that aimed for a better balance of costs and benefits: a "fast path" IP address and port matcher acted as a prefilter for the "slow path," a full HTTP proxy. The system, in use since 2004, was designed to block access to any of a secret list of pedophile web sites compiled by a third party. The author identifies ways to circumvent or attack such a system: use a proxy, use source routing to evade the blocking router, obfuscate requested URLs, use an alternate IP address or port, return false DNS results to put third parties on the "bad" list. They demonstrate that the two-level nature of the blocking system unintentionally makes it an oracle that can reveal the IP addresses of sites in the secret blocking list.

For a decade, the OpenNet Initiative produced reports on Internet filtering and surveillance in dozens of countries, until it ceased operation in 2014. For example, their 2005 report on Internet filtering in China [35] studied the problem from many perspectives, political, technical, and legal. They translated and interpreted Chinese laws relating to the Internet, which provide strong legal justifications for filtering. The laws regulate both Internet users and service providers, including cybercafes. They prohibit the transfer of information that is indecent, subversive, false, criminal, or that reveals state secrets. The OpenNet Initiative tested the extent of filtering of web sites, search engines, blogs, and email. They found a number of blocked web sites, some related to news and politics, and some on sensitive subjects such as Tibet and Taiwan. In some cases, entire sites (domains) were blocked; in others, only specific pages within a larger site were blocked. In a small

number of cases, sites were accessible by IP address but not by domain name. There were cases of overblocking: apparently inadvertently blocked sites that simply shared an IP address or URL keyword with an intentionally blocked site. On seeing a prohibited keyword, the firewall blocked connections by injecting a TCP RST packet to tear down the connection, then injecting a zero-sized TCP window, which would prevent any communication with the same server for a short time. Using technical tricks, the authors inferred that Chinese search engines index blocked sites (perhaps having a special exemption from the general firewall policy), but do not return them in search results. The firewall blocks access searches for certain keywords on Google as well as the Google Cache—but the latter could be worked around by tweaking the format of the URL. Censorship of blogs comprised keyword blocking by domestic blogging services, and blocking of external domains such as blogspot.com. Email filtering is done by the email providers themselves, not by an independent network firewall. Email providers seem to implement their filtering rules independently and inconsistently: messages were blocked by some providers and not others.

In 2006, Clayton, Murdoch, and Watson [9] further studied the technical aspects of the Great Firewall of China. They relied on an observation that the firewall was symmetric, treating incoming and outgoing traffic equally. By sending web requests from outside the firewall to a web server inside, they could provoke the same blocking behavior that someone on the inside would see. They sent HTTP requests containing forbidden keywords (e.g., "falun") caused the firewall to inject RST packets towards both the client and server. Simply ignoring RST packets (on both ends) rendered the blocking mostly ineffective. The injected packets had inconsistent TTLs and other anomalies that enabled their identification. Rudimentary countermeasures such as splitting keywords across packets were also effective in avoiding blocking. The authors of this paper bring up an important point that would become a major theme of future censorship modeling: censors are forced to trade blocking effectiveness against performance. In order to cope with high load at a reasonable costs, censors may choose the architecture of a network monitor or intrusion detection system, one that can passively monitor and inject packets, but cannot delay or drop them.

A nearly contemporary study by Wolfgarten [45] reproduced many of the results of Clayton, Murdoch, and Watson. Using a rented server in China, the author found cases of DNS tampering, search engine filtering, and RST injection caused by keyword sniffing. Not much later, in 2007, Lowe, Winters, and Marcus [32] did detailed experiments on DNS tampering in China. They tested about 1,600 recursive DNS servers in China against a

list of about 950 likely-censored domains. For about 400 domains, responses came back with bogus IP addresses, chosen from a set of about 20 distinct IP addresses. Eight of the bogus addresses were used more than the others: a whois lookup placed them in Australia, Canada, China, Hong Kong, and the U.S. By manipulating TTLs, the authors found that the false responses were injected by an intermediate router: the authentic response would be received as well, only later. A more comprehensive survey [4] of DNS tampering and injection occurred in 2014, giving remarkable insight into the internal structure of the censorship machines. DNS injection happens only at border routers. IP ID and TTL analysis show that each node is a cluster of several hundred processes that collectively inject censored responses. They found 174 bogus IP addresses, more than previously documented. They extracted a blacklist of about 15,000 keywords.

[46]

The Great Firewall, because of its unusual sophistication, has been an enduring object of study. Part of what makes it interesting is its many blocking modalities, both active and passive, proactive and reactive. The ConceptDoppler project of Crandall et al. [10] measured keyword filtering by the Great Firewall and showed how to discover new keywords automatically by latent semantic analysis, using the Chinese-language Wikipedia as a corpus. They found limited statefulness in the firewall: sending a naked HTTP request without a preceding SYN resulted in no blocking. In 2008 and 2009, Park and Crandall [36] further tested keyword filtering of HTTP responses. Injecting RST packets into responses is more difficult than doing the same to requests, because of the greater uncertainty in predicting TCP sequence numbers once a session is well underway. In fact, RST injection into responses was hit or miss, succeeding only 51% of the time, with some, apparently diurnal, variation. They also found inconsistencies in the statefulness of the firewall. Two of ten injection servers would react to a naked HTTP request; that it, one sent outside of an established TCP connection. The remaining eight of ten required an established TCP connection. Xu et al. [48] continued the theme of keyword filtering in 2011, with the goal of discovering where filters are located at the IP and AS levels. Most filtering is done at border networks (autonomous systems with at least one non-Chinese peer). In their measurements, the firewall was fully stateful: blocking was never triggered by an HTTP request outside an established TCP connection. Much filtering occurs at smaller regional providers, rather than on the network backbone.

Winter and Lindskog [44] did a formal investigation into active probing, a reported capability of the Great Firewall since around October 2011. They

focused on the firewall's probing of Tor relays. Using private Tor relays in Singapore, Sweden, and Russia, they provoked active probes by simulating Tor connections, collecting 3295 firewall scans over 17 days. Over half the scan came from a single IP address in China; the remainder seemingly came from ISP pools. Active probing is initiated every 15 minutes and each burst lasts for about 10 minutes.

Sfakianakis et al. [38] built CensMon, a system for testing web censorship using PlanetLab nodes as distributed measurement points. They ran the system for for 14 days in 2011 across 33 countries, testing about 5,000 unique URLs. They found 193 blocked domaincountry pairs, 176 of them in China. CensMon reports the mechanism of blocking. Across all nodes, it was 18.2% DNS tampering, 33.3% IP address blocking, and 48.5% HTTP keyword filtering. The system was not run on a continuing basis. Verkamp and Gupta [42] did a separate study in 11 countries, using a combination of PlanetLab nodes and the computers of volunteers. Censorship techniques vary across countries; for example, some show overt block pages and others do not. China was the only stateful censor of the 11.

PlanetLab is a system that was not originally designed for censorship measurement, that was later adapted for that purpose. Another recent example is RIPE Atlas, a globally distributed Internet measurement network consisting of physical probes hosted by volunteers, Atlas allows 4 types of measurements: ping, traceroute, DNS resolution, and X.509 certificate fetching. Anderson et al. [2] used Atlas to examine two case studies of censorship: Turkey's ban on social media sites in March 2014 and Russia's blocking of certain LiveJournal blogs in March 2014. In Turkey, they found at least six shifts in policy during two weeks of site blocking. They observed an escalation in blocking in Turkey: the authorities first poisoned DNS for twitter.com, then blocked the IP addresses of the Google public DNS servers, then finally blocked Twitter's IP addresses directly. In Russia, they found ten unique bogus IP addresses used to poison DNS.

Most research on censors has focused on the blocking of specific web sites and HTTP keywords. A few studies have looked at less discriminating forms of censorship: outright shutdowns and throttling without fully blocking. Dainotti et al. [11] reported on the total Internet shutdowns that took place in Egypt and Libya in the early months of 2011. They used multiple measurements to document the outages as they occurred: BGP data, a large network telescope, and active traceroutes. During outages, there was a drop in scanning traffic (mainly from the Conficker botnet) to their telescope. By comparing these different measurements, they showed that the shutdown in Libya was accomplished in more that one way, both by altering network

routes and by firewalls dropping packets. Anderson [1] documented network throttling in Iran, which occurred over two major periods between 2011 and 2012. Throttling degrades network access without totally blocking it, and is harder to detect than blocking. The author argues that a hallmark of throttling is a decrease in network throughput without an accompanying increase in latency and packet loss, distinguishing throttling from ordinary network congestion. Academic institutions were affected by throttling, but less so than other networks. Aryan et al. [5] tested censorship in Iran during the two months before the June 2013 presidential election. They found multiple blocking methods: HTTP request keyword filtering, DNS tampering, and throttling. The most usual method was HTTP request filtering. DNS tampering (directing to a blackhole IP address) affected only three domains: facebook.com, youtube.com, and plus.google.com. SSH connections were throttled down to about 15% of the link capacity, while randomized protocols were throttled almost down to zero 60 seconds into a connection's lifetime. Throttling seemed to be achieved by dropping packets, thereby forcing TCP's usual recovery.

Khattak et al. [30] evaluated the Great Firewall from the perspective that it works like an intrusion detection system or network monitor, and applied existing technique for evading a monitor the the problem of circumvention. They looked particularly for ways to evade detection that are expensive for the censor to remedy. They found that the firewall is stateful, but only in the client-to-server direction. The firewall is vulnerable to a variety of TCP- and HTTP-based evasion techniques, such as overlapping fragments, TTL-limited packets, and URL encodings.

Nabi [34] investigated web censorship in Pakistan in 2013, using a publicly known list of banned web sites. They tested on 5 different networks in Pakistan. Over half of the sites on the list were blocked by DNS tampering; less than 2% were additionally blocked by HTTP filtering (an injected redirection before April 2013, or a static block page after that). They conducted a small survey to find the most commonly used circumvention methods in Pakistan. The most used method was public VPNs, at 45% of respondents.

Ensafi et al. [15] employed an intriguing technique to measure censorship from many locations in China—a "hybrid idle scan." The hybrid idle scan allows one to test TCP connectivity between two Internet hosts, without needing to control either one. They selected roughly uniformly geographically distributed sites in China from which to measure connectivity to Tor relays, Tor directory authorities, and the web servers of popular Chinese web sites. There were frequent failures of the firewall resulting in temporary connectivity, typically lasting in bursts of hours.

In 2015, Marczak et al. [33] investigated an innovation in the capabilities of the border routers of China, an attack tool dubbed the "Great Cannon." The cannon was responsible for denial-of-service attacks on Amazon CloudFront and GitHub. The unwitting participants in the attack were web browsers located outside of China, who began their attack when the cannon injected malicious JavaScript into certain HTTP responses originating in China. The new attack tool is noteworthy because it demonstrated previously unseen in-path behavior, such as packet dropping.

Not every censor is China, with its sophisticated homegrown firewall. A major aspect of censor modeling is that many censors use commercial firewall hardware. A case in point is the analysis by Chaabane et al. [7] of 600 GB of leaked logs from Blue Coat proxies used for censorship in Syria. The logs cover 9 days in July and August 2011, and contain an entry for every HTTP request. The authors of the study found evidence of IP address blocking, domain name blocking, and HTTP request keyword blocking, and also of users circumventing censorship by downloading circumvention software or using the Google cache. All subdomains of .il, the top-level domain for Israel, were blocked, as were many IP address ranges in Israel. Blocked URL keywords included "proxy", "hotspotshield", "israel", and "ultrasurf" (resulting in collateral damage to the Google Toolbar and Facebook Like button because they have "proxy" in HTTP requests). Tor was only lightly censored—only one of several proxies blocked it, and only sporadically.

B Summary of circumvention systems

Many circumvention systems have been proposed or deployed. My survey with Tschantz, Afroz, and Paxson [40] covered 54 systems; a later one by Khattak, Elahi, et al. [28] covered 73. The systems mentioned in this section are not exhaustive but are chosen to be representative.

Against content blocking, circumvention systems generally take one of two strategies. The first is steganography, trying to blend in with some other protocol that the censor does not already block. The second is polymorphism, trying to look unlike anything the censor already blocks. Which one is more appropriate depends on the censor model. Against a censor that whitelists a small number of protocols and prohibits everything else, steganography is appropriate. Against a censor that blacklists a small number of protocols or keywords, polymorphism is appropriate. (The common understanding is that real-world censors tend to be of the blacklisting type, because whitelisting causes too much inherent collateral damage—it is too

hard to enumerate all the protocols users might want to use. The exception is in exceptionally constrained networks such as that of Cuba, that do not derive as much benefit from Internet connectivity anyway, and so can afford the collateral damage.)

FTE [13] (for "format-transforming encryption") is a quintessential example of a steganographic protocol. Given a specification of a regular expression, FTE transforms traffic to match it. The purpose is to force false-negative misclassification by firewalls. StegoTorus [43] uses custom encoders to make traffic resemble common HTTP file types, such as PDF, JavaScript, and Flash. FreeWave [23] modulates a data stream into an acoustic signal and transmits it over VoIP.

The history of the polymorphic, randomized protocols known as obfs2 [25], obfs3 [26], and obfs4 [3] is interesting because it tells a story of circumventors changing behavior in the face of changing censor models. All of these protocols aim to encode traffic as a uniformly random sequence of bytes, leaving no plaintext features for a censor to detect. The obfs2 protocol used a fairly naive handshake protocol that appeared random only to a first approximation. It would have bypassed the keyword- or pattern-based censors of its era, but it was detectable passively, using a custom detector. obfs3 improved on obfs2 by adding a clever Diffie–Hellman key exchange, specially modified to also appear random to a censor. obfs3 was not trivially detectable passively, but could be attacked by an active man in the middle, and was vulnerable to active probing. obfs4 added an out-of-band secret that foils both man-in-the-middle and active probing attacks.

"Decoy routing" systems put proxies at the middle of network paths. A special cooperating router lies between the client and the apparent destination of a TCP stream. The router looks for a special cryptographic "tag" that is undetectable to the censor. On finding a tag, the router begins to redirect the client's traffic away from its declared destination and towards a censored destination instead. There are several decoy routing proposals, each with advantages and disadvantages; those that began the line of research are called Curveball [27], Telex [47], and Cirripede [22].