Technical Report

UCAM-CL-TR-897
ISSN 1476-2986

Number 897

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

Characterization of Internet censorship from multiple perspectives

Sheharbano Khattak

January 2017

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted January 2017 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Robinson College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Summary

Internet censorship is rampant, both under the support of nation states and private actors, with important socio-economic and policy implications. Yet many issues around Internet censorship remain poorly understood because of the lack of adequate approaches to measure the phenomenon at scale. This thesis aims to help fill this gap by developing three methodologies to derive censorship ground truths, that are then applied to real-world datasets to study the effects of Internet censorship. These measurements are given foundation in a comprehensive taxonomy that captures the mechanics, scope, and dynamics of Internet censorship, complemented by a framework that is employed to systematize over 70 censorship resistance systems.

The first part of this dissertation analyzes *user-side censorship*, where a device near the user, such as the local ISP or the national backbone, blocks the user's online communication. This study provides quantified insights into how censorship affects users, content providers, and Internet Service Providers (ISPs); as seen through the lens of traffic datasets captured at an ISP in Pakistan over a period of three years, beginning in 2011.

The second part of this dissertation moves to *publisher-side censorship*. This is a new kind of blocking where the user's request arrives at the Web publisher, but the publisher (or something working on its behalf) refuses to respond based on some property of the user. Publisher-side censorship is explored in two contexts. The first is in the context of an anonymity network, Tor, involving a systematic enumeration and characterization of websites that treat Tor users differently from other users.

Continuing on the topic of publisher-side blocking, the second case study examines the Web's differential treatment of users of adblocking software. The rising popularity of adblockers in recent years poses a serious threat to the online advertising industry, prompting publishers to actively detect users of adblockers and subsequently block them or otherwise coerce them to disable the adblocker. This study presents a first characterization of such practices across the Alexa top 5K websites.

This dissertation demonstrates how the censor's blocking choices can leave behind a detectable pattern in network communications, that can be leveraged to establish exact mechanisms of censorship. This knowledge facilitates the characterization of censorship from different perspectives; uncovering entities involved in censorship and targets of censorship, and the effects of such practices on stakeholders. More broadly, this study complements efforts to illuminate the nature, scale, and effects of opaque filtering practices; equipping policy-makers with the knowledge necessary to systematically and effectively respond to Internet censorship.

To Shahmeer

for putting the colour inside of my world

Acknowledgements

This work has been shaped by the support and encouragement of a whole host of people, to whom I wish to express my profound gratitude.

I am grateful to Ali Khayam for instilling in me a passion for research and exploration. His perennial optimism and energy have been a source of great inspiration to me. He also played a key role in materializing our study of censorship in Pakistan.

The seeds of this dissertation were planted during my internship with Vern Paxson at The International Computer Science Institute (ICSI), Berkeley, in Autumn, 2012. I am fortunate to have been mentored by him since then. Vern patiently listened to many ideas and provided feedback on several drafts, teaching me along the way about a range of topics—from soundness and rigour in measurement studies, to effective management of collaborative research.

I am immensely grateful to Jon Crowcroft for all the freedom he gave me in pursuing my research interests. Jon's insightful comments on my research work have helped me to see the bigger picture. The consideration and respect he affords to his students has made it a privilege to work with him.

My heartfelt thanks to Steven Murdoch for helping me every step of the way. His feedback proved invaluable to shape and develop my research ideas into mature work. From Steven, I learnt going the extra mile to do things right; and the value of patient and consistent hard work.

I am deeply indebted to Ross Anderson for his unwaivering support and generous encouragement. The life lessons he imparted have equipped me with fundamental capabilities to cope with challenging situations.

I have greatly benefited from the support and mentorship of Mobin Javed; without her it would not have been the same. I am grateful to all my collaborators: Colleen Swanson, Damon McCoy, David Fifield, Emiliano De Cristofaro, Hamed Haddadi, Ian Goldberg, Julia E. Powles, Laurent Simon, Marjan Falahrastegar, Narseo Vallina-Rodriguez, Rishab Nithyanand, Sadia Afroz, Srikanth Sundaresan, Tariq Elahi, and Zartash Afzal Uzmi. I would like to thank a number of others who facilitated this work: Arturo Filastò, Bjoern A. Zeeb, Georg Koppen, George Danezis, Juris Vetra, Michael Tschantz, Moritz Bartl, Philipp Winter, and Zakir Durumeric. This work would not have been possible without the cooperation of the anonymous Internet Service Provider in Pakistan and the operators of the Tor exit nodes used in this study. I am grateful to the technical staff at the University of California, Berkeley, University of Cambridge, and University of Michigan for facilitating our scanning experiments. I thank the anonymous IMC, NDSS, IEEE S&P, PETS, and USENIX FOCI reviewers, and our shepherds, Olaf Maennel and Lujo Bauer, whose valuable feedback led to this work being more solid. I am grateful to my Ph.D. examiners, Robert Watson at the Computer Lab and Renata Cruz Teixeira at Inria Paris, for their thorough and constructive feedback; the final draft of this dissertation has

benefited greatly from their input.

I have been privileged to have had the opportunity to work with many brilliant and helpful people at the Computer Lab. Specifically, I thank Alice Hutchings, Dongting Yu, Ilias Marinos, Kumar Sharad, Richard Mortier, Robert Watson, Rubin Xu, and Sophie Van Der Zee. Special thanks to Julia E. Powles for providing useful comments on framing this work, and generally on effective writing. I am grateful to Caroline Stewart and Lise Gough at the Computer Lab, and Dr Julie Smith at Robinson College in helping me overcome several administrative hurdles that inevitably pop up during graduate years. Thanks to Mateja Jamnik and others involved in women@CL for creating a warm and supportive environment, and for consistently putting together exciting events. I am thankful to Aliya Khalid, Amna Abdul Wahid, Heidi Howard, Jyothish Soman, Mohibi Hussain, Negar Miralaei, Sharmeen Lodhi, and all others who gave freely of their time and friendship.

I am deeply grateful to have Jeunese Payne as my friend, who has been a continual source of support. Her kindness and understanding have brightened up many days. Warmest appreciation also to Farwa Bukhari for her relentless support. Finally, I thank my son Shahmeer for being my anchor always—this dissertation is dedicated to him.

This work was supported by the Engineering and Physical Sciences Research Council [grant number EP/L003406/1].

Contents

Chapter 1

Introduction

> "When *I* use a word," Humpty Dumpty said, in rather a scornful tone,
> "it means just what I choose it to mean—neither more nor less."
> "The question is," said Alice,
> "whether you *can* make words mean so many different things."
> "The question is," said Humpty Dumpty,
> "which is to be master—that's all."
>
> Lewis Carroll, "Through the Looking-Glass and What Alice Found There"

Censorship of online communications threatens principles of openness and freedom of information on which the Internet was founded. In the interest of transparency and accountability, and more broadly to develop scientific rigour in the field, we need methodologies to measure and characterize Internet censorship. Such studies will not only help users make informed choices about information access, but also illuminate entities involved in or affected by censorship; informing the development of policy and enquiries into the ethics and legality of such practices. However, measurement of Internet censorship is more complex than typical communication network measurements because of the inherently adversarial and opaque landscape in which it operates. As details about mechanisms and targets of censorship are usually undisclosed, it is hard to define exactly what comprises censorship, and how it operates in different contexts. This thesis aims to help fill this gap by developing three methodologies to characterize Internet censorship from multiple perspectives.

1.1 Background

Digital communication has emerged as a powerful medium for information dissemination, and as a facilitator of political freedom, social change and commerce. Consequently, various actors have resorted to control information that is considered undesirable—practices referred to as filtering, blocking, or censorship when conducted by a state. The goal of a censor is to disrupt free flow of information; potentially involving a range of steps to

stop the publication of information, to prevent access to information (e.g. by disrupting the link between the user and the publisher), or to directly prevent users from accessing information. While state-led filtering is more prevalent, there exist other actors that directly or indirectly control communications. Examples include a company launching denial of service attack on its competitor, Internet Service Providers (ISPs) offering to serve content from business partners only, or search engine providers manipulating search results by removing or downranking undesirable websites. (For consistency, through the rest of this dissertation we use the term 'censorship' to refer to all forms of communication control, and the term 'censor' to refer to the actor responsible for censorship, regardless of the state's involvement.) Such practices undermine freedom of expression—a basic human right endorsed by a large number of global organisations and governments, most notably by the US in the First Amendment [1], and by the United Nations in the Universal Declaration of Human Rights [2].

Motives of Internet Censorship. Driven by different goals, a diverse range of actors conduct censorship—for example, nation-states, industries, corporations, public facilities, and concerned parents seeking to protect their children from unsuitable Internet content. Precise enumeration of motives driving censorship is difficult, but generally fall under the themes of politics and power, social norms and morals, security, economics, and industrial goals [3] [4]. Politically motivated censorship is endemic to repressive regimes where the state controls information to serve its political agenda. The goal of social censorship is to control information that undermines accepted societal and moral values: typical targets include content related to pornography, homosexuality, gambling, hate speech, and criticism of those in power. Censorship can be driven by the desire to thwart security-related threats ranging from terrorism and insurgency, to malware, phishing, and spam. Protection of economic interests is another impetus for censorship: a number of countries filter foreign Internet services and platforms to boost local markets. Censorship is also conducted to protect industrial and business goals; for example, protection of intellectual property rights, and issues concerning net neutrality where ISPs discriminate against content or content providers through preferential treatment. Internet services, protocols, and tools that support the flow of objectionable information (e.g. circumvention tools, translation services, email-providers, and communication platforms like micro-blogging websites and Web hosting applications) are also common targets of censorship.

Origins of Censorship. Censorship is not a recent phenomenon. Throughout history, those in power have sought to control flows of information to maintain their authority [5]. Such practices can be traced to the office of censor in Rome as early as 443 BC, established to shape character of the people. China saw its first censorship law in 300 AD. As orthodoxy became established in Europe, the church increasingly perceived free speech as a threat to Christian doctrine, resorting to censorship. The invention of printing press in

Europe in the mid 15th century led the church to further tighten its grips over information, extending its control to all universities. Throughout Europe, the church and state formed a close alliance to control the publication and sale of books, the effect of which permeated to the colonised territories in the Americas. The development of printing press in Europe instigated publication of newsletters and newspapers—the first newspaper appearing in 1610 in Switzerland, soon followed by other European countries. The growing popularity of newspapers gave rise to concerns about its effect on social and moral values, especially during times of political unrest. The press was thus heavily censored and regulated in Britain in 1662–1665, and in Germany in 1618–48. The postal service, first established in France in 1464, also became a common target of censorship because of its key role in facilitating communication, especially in times of war. Similarly, libraries have historically been the target of censorship—the first recorded incident of burning of a library took place in China as far back as 221 BC.

Internet Censorship: A Historical Overview. The modern era is marked by the Internet revolution which changed the way we perceive and conduct communications in an unprecedented way. In 1969, a research project titled ARPAnet (by Advanced Research Projects Agency, a division of the US Department of Defense) laid groundwork for the Internet. By 1995, the Internet could be used to carry commercial traffic without any restrictions, emerging as a powerful medium of communication—carrying 51% of the information flowing through two-way telecommunications networks in 2000, and 97% by 2007 [6]. As a result, governments and other actors turned to enforce censorship in cyberspace by regulating and filtering different parts of the Internet. The OpenNet Initiative (ONI) breaks down the history of Internet censorship into four phases [7]:

- **The Open Commons (1960–2000).** During the period from the Internet's initial development in 1960s to about 2000, the Internet was seen as a space separate than reality that was subject to little or no authoritative regulation. The Internet facilitated access to information at a massive scale, and enabled global communication at a low cost. It also emerged as an effective vehicle for democratization as individuals formed forums and online communities for collective action.

- **Access Denied (2000–2005).** During this period, states such as China and Saudi Arabia and other actors began to see the Internet as something that needs to be regulated. Censorship typically took the form of blocking access to IP addresses, hosts, and domains. China deployed a sophisticated and comprehensive censorship system, proving that a motivated and resourceful government can institute traditional controls in cyberspace. In addition to historically authoritative governments, democratic states also turned to regulate the Internet—child pornography being the most pervasive target of filtering.

- **Access Controlled (2005–2010).** In this phase, the censors moved from direct blocking to more flexible filters that could be tuned according to changes in political and social events, including aggressive controls such as denial of service attacks and cyber espionage. Some states mandated registration, licensing, and identification to access online resources, creating an environment of self-censorship. A number of private companies emerged to provide censorship technology and expertise to 'client' states. Another trend was to use legal controls to delegate censorship to key actors such as search engines, cloud-computing services, access points, and hosting platforms. Even in the absence of state intervention, some of these intermediary actors made arbitrary and at times discriminatory business decisions. In China, the so-called 'Fifty Cent Party' (named after the amount of money per Internet post purportedly paid to its members) was developed to shape public opinion on chat rooms, blogs, and online forums by posting comments that glorified the regime and distracted from its criticism.

- **Access Contested (2010–Present).** The current phase is characterized by an overt tussle between various actors including advocates for free speech, and governments and corporations to achieve their respective goals. Cyberspace is growing increasingly militarized, where states actively engage in information-warfare. Companies that were being pressured by the state in the previous phase to comply with local censorship laws are developing strategies such as the Global Network Initiative [8] to respond to filtering requests in a manner that does not conflict with freedom of speech. Free speech advocates are assertively resisting filtering policies with success in a number of cases. The contests between actors in the cyberspace with competing goals has given rise to an arms race.

1.2 Motivation

As the technology that is used to retrieve and disseminate information evolves, mechanisms of censorship become more sophisticated in tandem. A recent report by Freedom House estimates that more than 60 countries around the world engage in some form of censorship [9]. The problem is expected to exacerbate as Internet adoption around the world grows. Of yet greater concern is the fact that we have little visibility into censorship; because of the inherent sensitivity of the subject, practices largely remain discreet, potentially with the support of various actors in addition to conventional state-level censors.

While censorship deployment and technology is an active field of research and scholarship, there is still a pressing need for a broad perspective on how censorship operates and how it affects the Internet ecosystem of users and publishers, and between them a range of stakeholders such as service providers and content providers. Illumination of these issues has important social, economic, ethical, legal, and policy implications. However, this has remained largely unexplored because measuring censorship presents

four challenges that make the task more involved than measurement of regular network phenomena such as performance. First, censorship inherently operates in an adversarial environment; as a result of which key contextual information—*what* was censored, and *how* it was censored—is completely or partially missing, making it difficult to define exactly what comprises censorship, and what forms the ground truth against which to benchmark measurements. Second, the metrics used to characterize censorship need to be tailored on a case-by-case basis as the targets and mechanisms of censorship vary over time and across different jurisdictions. Third, studying broad effects of Internet censorship requires access to datasets collected at appropriate vantage points, which may be logistically infeasible. Finally, the measurement process must incorporate consideration of incidental failures such as those caused by packet loss or network outages. Additionally, measurements must incorporate subtleties so as not to trigger the censorship apparatus causing measurement bias analogous to the Heisenberg effect. For example, a common methodology for detecting censorship is to send requests to a list of websites from a test location and a control location: websites that are accessible only from the control location are potentially censored at the test site. Though apparently straightforward, this approach can potentially generate false positives. Some censors, once triggered, continue to block all subsequent requests from the source IP address for some time (e.g. the Great Firewall of China blocks a source IP address for 90 seconds after observing objectionable activity from it [10]). Thus, consecutive requests must be spaced far enough apart to allow the censor to cool off.

1.3 Research Question and its Substantiation

This dissertation focuses on forms of censorship that involve blocking, arguing the following thesis:

> *The censor has an array of choices for how to block, some of which leave a trail in network traces. The presence of such a trail—a sequence of packets (not necessarily contiguous), or an absence of expected packets)—at suitable vantage points and after removal of ambiguities and inaccuracies (sanitization), implies censorship. The defined mechanisms of censorship can be leveraged to characterize various aspects of censorship—identifying entities involved in censorship, what content is censored, and the effects on intermediate players.*

This thesis is substantiated in two threads of research. The first part studies *user-side censorship*: the publisher is willing to accept a user's connection, but some device along the path between the user and the publisher blocks it. Using network logs captured at an ISP in Pakistan during a period of escalating censorship, this study develops methodologies to reconstruct censorship ground truth through passive analysis of network data, and investigates consequences of Internet censorship on end users, service providers, and content providers. The second part of this study shifts focus to *publisher-side censorship*: the user's

connection arrives at the publisher unimpeded, but the publisher (or something working on its behalf) rejects it based on some characteristics of the source. This phenomenon is examined in two contexts: *(i)* an anonymity network, Tor, which often provides the means for citizens to access or distribute censored or restricted content while protecting their privacy or even safety, and *(ii)* adblocking software that users install to disable online ads, mainly to improve their Web browsing experience while maintaining their privacy. This work produced methodologies for active measurement of the Web's blocking of certain classes of users, mapping out the entities that are directly or indirectly involved in such practices. Additionally, given the breadth of this work and to support the conclusions made, this dissertation includes a comprehensive censor's attack model, and a framework to evaluate over 70 censorship resistance systems.

1.4 Dissertation Organization and Contributions

This dissertation makes the following contributions:

- **Systematization of Knowledge of Internet Censorship and Censorship Resistance Systems.** Developing methodologies to detect censorship and studying user behaviour with respect to circumventing censorship requires a solid understanding of these two domains. However, the censor's attack landscape has become significantly complicated because of the ongoing arms race between censors and the systems used to circumvent them. This dissertation *presents a comprehensive model that captures how censorship takes place*, the scope of censorship, and the dynamics that influence it; and *a framework to systematize over 70 existing censorship resistance systems by their threat models and corresponding defenses.* (Chapter 2)

- **Characterization of How Censorship Affects End Users, Service Providers, and Content Providers.** Often data relevant to censorship (e.g. leaked network logs) becomes available long after the incident, requiring forensic analysis to uncover what was censored and how it was censored. This dissertation *develops a methodology to reconstruct censorship ground truth through passive analysis of data* by using censorship indicators: analyzing the responses seen from servers in reply to user requests, basing deductions on the observation that for enforcing censorship, a censor either silently drops requests or sends back false response packets. These methodologies are applied to two large-scale censorship events in Pakistan, blocking of pornographic content in 2011 and of YouTube in 2012, to *analyze how censorship affects a range of stakeholders that includes end users, service providers, and content providers.* Using traffic datasets collected at an ISP in Pakistan before and after the censorship events, this study: *(i)* quantifies the demand for blocked content, *(ii)* illuminates challenges encountered by service providers in implementing the censorship policies, *(iii)* investigates changes in user behaviour (e.g. with respect to

circumvention) after censorship, and *(iv)* assesses benefits extracted by competing content providers of blocked content. (Chapter 3)

- **Characterization of Differential Treatment of Tor Anonymity Network Traffic.** Recent years have seen the rise of publisher-side censorship. Unlike the common form of censorship where blocking takes place near the user (e.g. state-level censorship where an intermediate device such as a border router drops user requests for blacklisted websites), in publisher-side blocking the user's request arrives at the publisher, but the publisher (or something working on its behalf) refuses to respond based on some property of the user. So far, the efforts to record such blocking have been ad hoc; effectively by cataloging reports from frustrated users about services that routinely employ such practices. This dissertation *develops a methodology to comprehensively enumerate the Web's differential treatment of a certain class of users* by examining blocking at the transport layer through reset or dropped connections, and at the application layer, through explicit blocks served from website home pages. The defined methodology is employed to analyze publisher-side censorship of an anonymity network, Tor, that is often employed by citizens in repressed regimes to privately access or distribute censored content without a threat to their safety. The results help to *map out the websites that block Tor, providing insights into the nature of this differential treatment*; that is, how much of it is because of explicit decisions to block Tor versus the consequence of fate-sharing because of automated abuse-based blocking. (Chapter 4)

- **Characterization of Differential Treatment of Users of Adblocking Software.** Adblocking tools like Adblock Plus continue to rise in popularity, potentially threatening the dynamics of advertising revenue streams. In response, a number of publishers have ramped up efforts to develop and deploy mechanisms for detecting or counter-blocking adblockers (referred to as anti-adblockers), effectively escalating the online advertising arms race. This dissertation *develops a scalable approach for identifying third-party services shared across multiple websites.* This methodology is used to provide *a first characterization of anti-adblocking across the Alexa top 5K websites*; mapping websites that perform anti-adblocking and the entities that provide anti-adblocking scripts, and sketching the modus operandi of these scripts and their interaction with popular adblockers. (Chapter 5)

This dissertation is organized as follows. Chapter 2 lays the foundation for measurements conducted in later parts of this dissertation by setting out a comprehensive taxonomy of Internet censorship and the systems that are used to circumvent such blocking. Chapter 3 presents an investigation of how state-level censorship affects various stakeholders (i.e. users, content providers, and service providers), providing insights into its social and economic impact. The next two chapters shift focus to a new kind of blocking where publishers block users of certain software or services: the Tor anonymity

network in Chapter 4, and adblocking software in Chapter 5. Chapter 6 concludes the dissertation by summarizing the key insights and avenues for future research. Additionally, each chapter includes a section on related work to provide a more focused comparison of the contributions of this dissertation with relevant previous work. Finally, a note on style: this dissertation employs an impersonal style in Chapters 1 and 6, but the core Chapters 2, 3, 4, and 5 use the personal pronoun *we* as these were produced through teamwork (as stated in Section 1.5), even though the author led the work (except for that described in Section 1.6).

1.5 Published Work

This dissertation has resulted in the following publications in peer reviewed academic conferences and workshops (in chronological order, relevant chapter indicated in bold):

- Rishab Nithyanand, Sheharbano Khattak, Mobin Javed, Narseo Vallina-Rodriguez, Marjan Falahrastegar, Julia E. Powles, Emiliano De Cristofaro, Hamed Haddadi and Steven J. Murdoch. Adblocking and Counter Blocking: A Slice of the Arms Race. Proceedings of the 6th USENIX Workshop on Free and Open Communications on the Internet (FOCI), 2016. **Chapter 5**.

- Sheharbano Khattak, Tariq Elahi, Laurent Simon, Colleen Swanson, Steven J. Murdoch and Ian Goldberg. SoK: Making Sense of Censorship Resistance Systems. Proceedings on Privacy Enhancing Technologies, Vol. 2016, No. 4 (PETS), 2016. **Chapter 2**.

- 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. Proceedings of the 23rd Network and Distributed System Security Symposium (NDSS), 2016. **Chapter 4**.

- Sheharbano Khattak, Mobin Javed, Syed Ali Khayam, Zartash Afzal Uzmi and Vern Paxson. A Look at the Consequences of Internet Censorship Through an ISP Lens. Proceedings of the 14th ACM SIGCOMM conference on Internet measurement (IMC), 2014. **Chapter 3**.

Listed below are other publications outside scope for inclusion:

- Sheharbano Khattak, Zaafar Ahmed, Affan A. Syed and Syed Ali Khayam. BotFlex: A community-driven tool for botnet detection. Elsevier Journal of Network and Computer Applications, Volume 58, December 2015, Pages 144–154.

- Sheharbano Khattak, Laurent Simon, and Steven J. Murdoch. Systemization of Pluggable Transports for Censorship Resistance, arXiv preprint, December 2014. *Note: The censorship attack model in this work largely overlaps with Section 2.1.4.*

1.6 Work Done in Collaboration

A large part of this work has been conducted in collaboration with other researchers. All the coauthors contributed to high-level development of the work listed in Section 1.5. Specifically, Tariq Elahi did the work on censorship distinguishers (Section 2.1.1), scope of censorship (Section 2.1.2), and an abstract model of censorship (Section 2.1.3) in Chapter 2. In the work on publisher-side blocking of Tor, Sadia Afroz and David Fifield worked on application layer discrimination (Section 4.4). In the work on blocking of users of adblocking software presented in Chapter 5, Rishab Nithyanand and Narseo Vallina-Rodriguez developed the mechanism for detecting third-party services shared across websites (Section 5.2).

Chapter 2

Internet Censorship and Censorship Resistance

In this chapter, we provide a broad perspective on capabilities of censors, approaches to censorship resistance, and the overarching trends and gaps in the field. An increasing number of countries implement Internet censorship at different scales and for a variety of reasons. As a result, censorship resistance systems (CRSes) have emerged to help bypass such blocks. Because of the diversity of censorship mechanisms across different jurisdictions and their evolution over time, there is no one approach which is optimally efficient and resistant to all censors. Consequently, an arms race has developed, resulting in the evolution of censorship resistance systems to have dramatically sped up. We conduct a comprehensive survey of 73 CRSes, including both deployed systems and those described in academic literature. We consolidate the threat models employed in the surveyed CRSes into a censorship attack landscape, with a discussion on the factors that can potentially affect policy (Section 2.1). Next, we categorize censorship resistance schemes based on recurring themes and underlying concepts in our CRS survey. We create a framework to describe censorship resistance in terms of security, privacy, performance, and deployability (Section 2.3); and evaluate CRS schemes, discussing their strengths and limitations (Sections 2.4 and 2.5). The chapter concludes by laying out a set of overarching research gaps and challenges to inform future research (Section 2.7).

2.1 Internet Censorship

The goal of a censorship resistance system (CRS) is to enable information exchange between users and publishers despite the censor's attempts to block the communication. Blocking might involve a range of steps: attacking the information itself (through corruption, insertion of false information, deletion, or modification), or by impairing information publication or access (Figure 2.1).

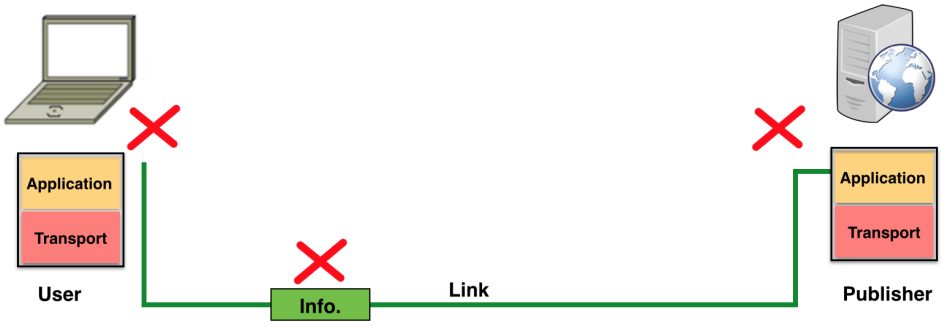


Figure 2.1: A system meant for information exchange involves a *user* that retrieves *information* from a *publisher* over a link that connects the two. The censor's goal is to disrupt information exchange either by corrupting it, or by hindering its access or publication (indicated by red crosses in the figure).

2.1.1 Censorship Distinguishers

Censorship is aided by *distinguishers* that are composed of feature–value pairs: A *feature* is an attribute (e.g. an IP address, protocol signature, or packet size distribution) with an associated *value* that can be a singleton, a list, or a range. Values are typically specified by a distribution over the set of all possible values that the feature can take. Where this distribution is sufficiently distinctive, feature–value pairs can be used as distinguishers to detect prohibited activities. For example, a censor can use the prevalence of 586-byte packet lengths in traffic as a distinguisher to identify Tor traffic. Distinguishers are high- or low-quality depending on whether these admit low or high error-rates, respectively. Furthermore, distinguishers can be low- or high-cost depending on whether small or large amounts of resources are required to utilize them. For the censor, high-quality–low-cost distinguishers are ideal. The primary source of distinguishers is network traffic between users and publishers, where feature–value pairs may correspond to headers and payloads of protocols at different network layers (e.g. source and destination addresses in IP headers, destination ports and sequence numbers in the TCP header, or TLS record content type in the TLS header). Unencrypted packet payloads can also reveal forbidden keywords. The censor can derive distinguishers from traffic statistics such as packet lengths and timing distributions. Additionally, the censor can use the previously described distinguishers to develop models of allowed traffic and disrupt anomalous flows.

2.1.2 Scope of Censorship

Censors vary widely with respect to their motivation, effectiveness, and technical sophistication. A wide range of entities, from individuals to corporations and state-level actors, may act as a censor. The extent to which a censor can effectively disrupt communication is a consequence of the censor's resources and constraints. Specifically, the censor's technical resources, capabilities, and goals are informed by its *sphere of influence* and *sphere of*

visibility. The sphere of influence is the degree of *active* control the censor has over the flow of information and behaviour of individuals or large entities. The sphere of visibility is the degree of *passive* visibility a censor has over the flow of information on its own networks and those of other operators.

The spheres of influence and visibility are dictated by *physical*, *political*, or *economic* dynamics. Limitations due to geography are an example of physical constraints. Relevant legal doctrine or international agreements and understandings that influence the censor's actions are examples of political limitations. Economic constraints assume that the censor operates within some specified budget that affects the technical sophistication and accuracy of the censorship apparatus it can field.

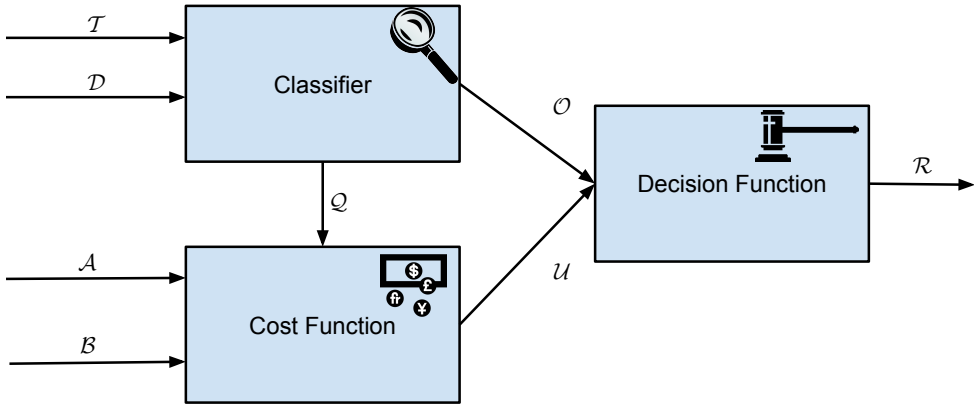


Figure 2.2: An abstract model of censorship. The classifier takes as inputs the set of network traffic to be analyzed, $\mathcal{T}$, and the set of distinguishers, $\mathcal{D}$, and outputs a set $\mathcal{O}$ of traffic labels and the associated false negative rate (FNR) and false positive rate (FPR), denoted by $\mathcal{Q}$. (The censor's FNR refers to the rate at which offending traffic is mislabeled as legitimate, and the FPR is the rate at which legitimate traffic is mislabeled as offending.) 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 $\mathcal{R}$.

2.1.3 An Abstract Model of Censorship

At an abstract level, the censorship apparatus is composed of classifier and cost functions that feed into a decision function (Figure 2.2). Censorship activity can be categorized into two distinct phases, *fingerprinting* and *direct censorship*.

In the first phase (fingerprinting), the censor identifies and then uses a set of distinguishers $\mathcal{D}$ to flag prohibited network activity. For example, the censor may employ regular expressions to detect flows corresponding to a blocked publisher. The classifier takes $\mathcal{D}$ and the set of network traffic to be analyzed $\mathcal{T}$ as inputs, and outputs offending

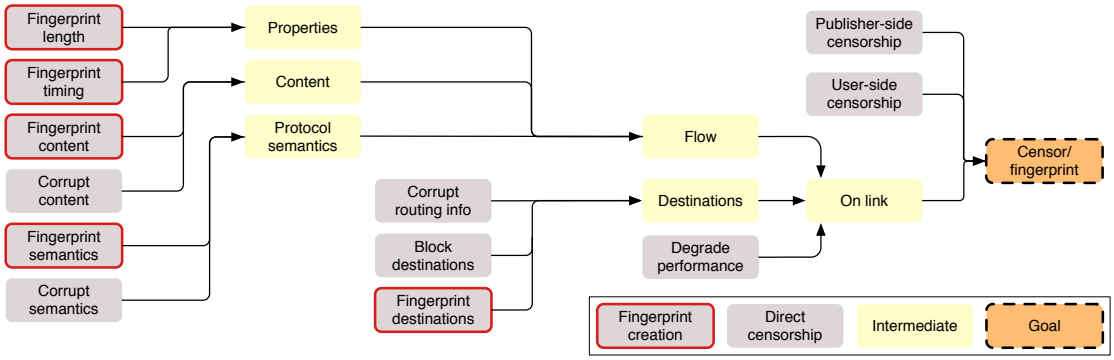


Figure 2.3: Censor's attack model, showing both direct censorship (information corruption, or disabling access or publication) and fingerprinting (to develop and improve features for direct censorship). This figure does not include the relatively intangible attacks of coercion, denial of service, and installation of censorship software on machines.

traffic flows within some acceptable margin of error to account for misclassification. The censor's error rates are the *false negative rate (FNR)* (or the *information leakage*), the rate at which offending traffic is mislabeled as legitimate; and the *false positive rate (FPR)*, the rate at which legitimate traffic is mislabeled as offending (causing *collateral damage*). We assume that the censor wants to minimize the FNR and FPR within its given set of constraints.

In the second phase (direct censorship), the censor responds to flagged network flows based on a utility function that accounts for the censor's costs and tolerance for errors. For example, the censor may choose to block flagged network flows by sending TCP reset packets to both the user and the publisher to force the connection to terminate.

2.1.4 Censor's Attack Model

We survey 73 CRSes, including deployed tools and academic papers (Appendix A). We consolidate the threat models sketched in these systems into a single attack model (Figure 2.3), expanding the concepts of fingerprinting and direct censorship described in the abstract model of censorship (Section 2.1.3). We group censorship activities by where these take place; that is, the user, the publisher, or the link between them.

Fingerprinting

Fingerprint Destinations. A flow can be associated with a protocol based on distinguishers derived from the connection tuple. Destination ports are a typical target of censorship (e.g. 80 for HTTP); and so are flows addressed to IP addresses, hosts, and domain names known to be associated with a blocked system. Flow fingerprinting can be used to constitute a multi-stage censorship policy, possibly followed by a blocking step. For example, a British ISP BT employed a hybrid two-stage censorship system (CleanFeed)

that first redirects traffic matching its IP blacklist to a HTTP proxy, and then performs content filtering on the redirected traffic [11].

Fingerprint Content. Flows can be fingerprinted by checking for the presence of protocol-specific strings, blacklisted keywords, domain names, and HTTP hosts. A number of deep packet inspection (DPI) systems can perform regex-based traffic classification [12] [13] [14] [15]; however, it remains unclear what are the true costs of performing DPI at scale [16] [17]. Alternatively, flows can be fingerprinted based on some property of the content being carried. For example, if the censor does not allow encrypted content, it can use high content entropy as a distinguisher to flag encrypted flows [18].

Fingerprint Flow Properties. The censor can fingerprint a protocol by creating its statistical model based on flow-based distinguishers (e.g. packet length) and timing-related features (e.g. inter-arrival times and burstiness) [19] [20]. Wiley [21] used Bayesian models created from sample traffic to fingerprint obfuscated protocols (Dust [21], SSL, and obfs-openssh [22]) based on flow features. An entropy-based detector was found to be more efficient (94% accurate using only the first packet) than length- and timing-based detectors (accuracy of 16% and 89%, respectively, over entire packet streams). Transport layer behaviour such as the number of outgoing connections can also be used for fingerprinting applications. Previous work has demonstrated the feasibility of flow properties in fingerprinting the websites visited by a user even if the flow is encrypted [23] [24] [25] [26].

Fingerprint Protocol Semantics. The censor can fingerprint flows based on protocol behaviour elicited through active manipulation (i.e. by dropping, injecting, modifying, and delaying packets). The censor's goal is to leverage knowledge of a protocol's semantic properties to tease out behaviour of a known protocol. Alternatively, the censor can conduct multiple fingerprinting steps to collect information on which to base subsequent blocking decisions. Wilde [27] found that the Great Firewall of China (GFW) employed a two-step process to block Tor bridges in 2011: *(i)* when a Tor client within China connects to a Tor bridge or relay, GFW's DPI box flags the flow as a potential Tor flow, and *(ii)* random Chinese IP addresses then try to establish a Tor connection with the bridge, resulting in the bridge getting blocked if the previous connection attempt was successful.

Direct Censorship

User-Side Censorship. The censor can directly or discreetly (facilitated by malware or insider attacks) install *censorship software* on user machines. This software can then disrupt information access; for example, by preventing installation of unapproved software, by disrupting functionality of Internet searches by returning pruned results, and by displaying warnings to dissuade users from attempting to find, distribute, or use censored content. Such blocking may lead to corruption of information as well as access disruption. China's

Green Dam, a filtering software purported to protect children from harmful Internet content, was mandated to be installed on all new Chinese computers in 2009 [28]. The software was found to be far more intrusive than officially portrayed, blocking access to a large blacklist of websites in diverse categories; moreover, it monitored and disrupted operation of other programs running on the same machine as itself if these were used to access censored content. TOM-Skype, a joint venture between a Chinese telephony company TOM Online and Skype Limited, is a Voice-over-IP (VoIP) chat client program that uses a list of keywords to censor chat messages in either direction [29].

The censor can *coerce* users who attempt to access blocked content. The censor can set up malicious resources such as proxy nodes or fraudulent documents, to attract unwary users, that counteract the publisher's goals. 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 [30], [31], [32], [33]. China 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 [34], who may then be targeted for punishment [35].

Publisher-Side Censorship. The censor can install *censorship software* on the publisher or employ a manual process to corrupt the information being published or disrupt the publication process. A number of studies investigate the Chinese government's censorship of posts on the national microblogging site Sina Weibo. Bamman *et al.* analyze three months of Weibo data and find that 16% of politically-driven content ends up getting deleted [36]. Zhu *et al.* note that Weibo's user-generated content is mainly removed during the hour following the post with about 30% of removals occurring within 30 minutes and about 90% within 24 hours [34]. Another study observes posts from politically active Weibo users over 44 days and finds that censorship varies across topics, with the highest deletion rate culminating at 82%. They further note the use of *morphs*—adapted variants of words to avoid keyword-based censorship. Weiboscope is a data collection, image aggregation, and visualization tool: it makes censored Sina Weibo posts by a set of Chinese microbloggers publicly available [37].

Within its sphere of influence, the censor can use legal or extralegal *coercion* to shut down a publisher. For example, in 2005, a popular anonymous email service was pressured by the U.S. government to reveal private user information [38]. The censor can coerce publishers into retracting publications; for example, through threats of imprisonment.

To censor destinations outside its sphere of influence, the censor can mount a network attack. For example, China launched an effective *denial of service (DoS)* attack against GitHub, targeting censorship resistance content hosted on the website [39].

Degrade Performance. The censor can manipulate characteristics of the link between users and publishers (e.g. by introducing delays and low connection time-out values); effectively disrupting CRSes that are not resilient to errors. Compared to more drastic

measures like severing network flows, degradation is a soft form of censorship that diminishes access to information while also affording deniability to the censor. A study of network quality (i.e. network congestion, packet loss, and latency) in Iran during the period 2010–2013 [40] reveals two extended periods of Internet throttling (with a 77% and 69% decrease in download throughput, respectively). These periods often coincided with holidays, protest events, international political turmoils, and important anniversaries; sometimes accompanied by overt filtering of online services or jamming of international broadcast television.

Block Destinations. To prevent information access or publication, the censor can leverage distinguishers derived from a connection tuple (source IP, source port, destination IP, and destination port), or other identifiers such as hosts and domain names. The block can continue for a short period of time to create a chilling effect and encourage self-censorship. One study notes that GFW blocks communication from a source IP address to a destination IP–port for 90 seconds after observing objectionable activity over that flow [10]. GFW has been reported to drop packets originating from Tor bridges based on both source IP address and source port to minimize collateral damage [41].

Corrupt Routing Information. The censor can disrupt access by corrupting information that helps in finding destinations on the Internet; for example, by changing routing entries on an intermediate censor-controlled router, or by manipulating information that supports the routing process. Border Gateway Protocol (BGP) is the de facto protocol for inter-AS routing. The censor can block a network's connectivity to the Internet by withdrawing previously advertised network prefixes, or by re-advertising them with different properties (rogue BGP route advertisements). A number of countries have enforced complete or partial Internet outages in recent years by withdrawing their networks from the Internet's global routing table (Egypt [42], Libya [43], Sudan [44] and Myanmar [45]). DNS is another vital service that maps names given to different Internet resources to IP addresses. The hierarchical distributed nature of DNS makes it vulnerable to censorship. Typical forms of DNS manipulation involve redirecting DNS queries for blacklisted domain names to a censor-controlled IP address (DNS redirection or poisoning), a non-existent IP address (DNS blackholing), or by simply dropping DNS responses for blacklisted domains. China was found to inject forged DNS responses to queries for blocked domain names, causing large-scale collateral damage when the same policy affected outside traffic traversing Chinese links [46].

Corrupt Flow Content. The censor can compromise information or disrupt access by corrupting transport layer payload of a flow. For example,the censor can inject a HTTP `404 Not Found` message in response to requests for censored content and drop the original response, or modify the HTML page in the body of an HTTP response.

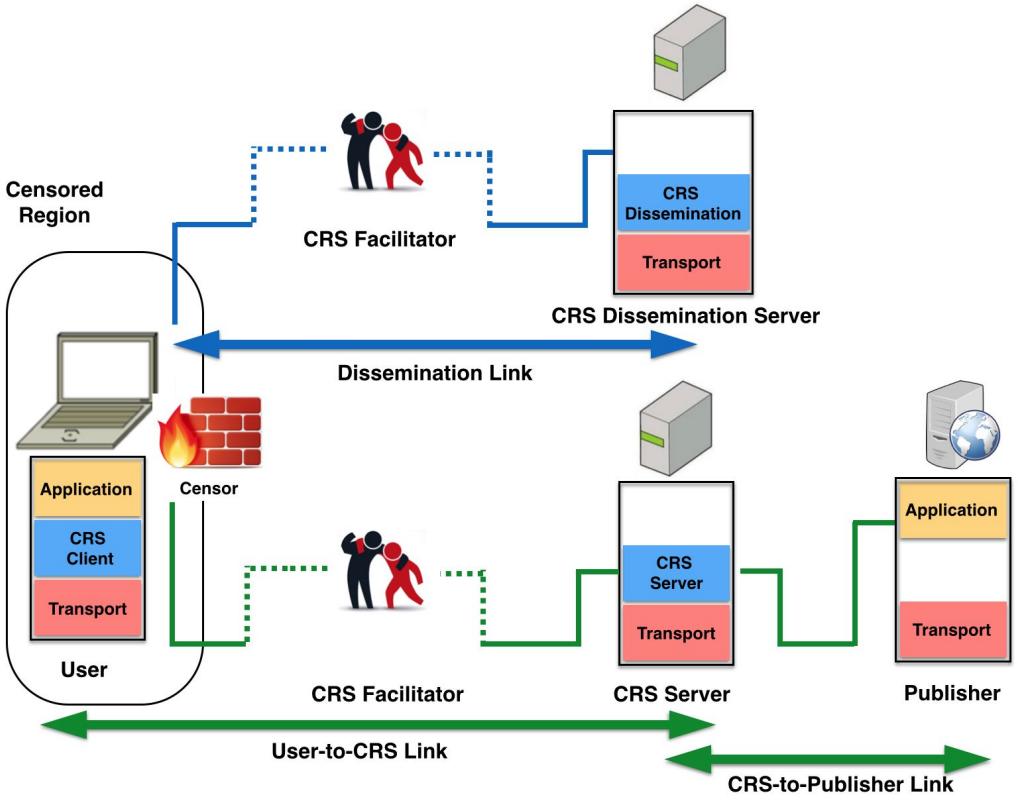


Figure 2.4: The Censorship Resistance System (CRS) provides users unfettered access to information despite censorship. The *CRS client* software (installed on the user's machine) obtains information (CRS credentials) about how to access the *CRS server* from the *dissemination server* over the *dissemination link*. The CRS client then uses these credentials to connect to the CRS server over the *user-to-CRS link*. Optionally, there could be one or more *CRS facilitators* on the dissemination link and the user-to-CRS link which support the CRS's operation. The CRS server connects to the publisher over the *CRS-to-publisher link*, effectively acting as a proxy that enables blocking-resistant communication between the user and the publisher. It is common for CRSes to handle the CRS-to-publisher link separately from the other two links because of their disparate security and performance requirements.

Corrupt Protocol Semantics. The censor can corrupt information or disrupt access by manipulating protocol semantics to induce disruption on a flow. For example, injecting forged TCP reset packets into a flow will cause both endpoints to tear down the connection.

In the next section we turn to CRSes. Note that there is not much that a CRS can do if the censor has installed malicious software on the user or publisher machines; we consider such attacks out-of-scope and do not discuss these in the rest of this chapter.

2.2 Censorship Resistance

A censorship resistance system (CRS) thwarts the censor's attempts to corrupt information, or to disrupt retrieval or publication of information. Censorship resistance typically involves

overcoming the censor's sphere of influence and sphere of visibility while maintaining an acceptable level of security and performance for CRS users. A CRS may be employed simultaneously for multiple *use cases* (e.g. whistleblowing, publishing information, and to organize strikes); each use case potentially requiring different properties of the CRS.

A CRS involves interaction between various components to enable unblockable communication between users and publishers (Figure 2.4). The user first installs the *CRS client* software that provides the desired CRS functionality. At a high level, we can break down CRS functionality into two distinct phases, *Communication Establishment* and *Conversation*.

Communication Establishment. Communication Establishment includes steps taken by the CRS client, from obtaining CRS credentials from the *dissemination server* to being able to access and use the *CRS server*; such that legitimate CRS users can easily learn CRS credentials, but the censor cannot harvest them efficiently. Additionally, the communication exchanges between the CRS client, and the dissemination and CRS servers should be resistant to fingerprinting.

As a preliminary step, the user may optionally employ an *out-of-band link* to gather bootstrapping information such as a secret key or token. The CRS uses the out-of-band link to obtain secrets that are assumed to somehow arrive with absolute security. An example of such a link is a person from outside the censor's sphere of visibility who brings addresses of CRS proxies on a USB stick through the airport and hand delivers it to the user.

The CRS client uses the dissemination link to connect to the dissemination server to retrieve information (*CRS credentials*) about how to join the CRS. CRS credentials may be public, such as domain name mappings to IP addresses or routing information; additionally, restricted CRS-specific information such as addresses of proxy servers or locations of censored content might also be required. Next the CRS client uses the CRS credentials previously obtained to connect to the CRS server over the user-to-CRS link. Some systems may directly connect to the CRS server, skipping the connection between the CRS client and the dissemination server. The CRS-client–dissemination-server and the CRS-client–CRS-server connections can optionally be facilitated by intermediate *participants* (e.g. proxies that relay traffic between the CRS client, and the dissemination and CRS servers).

Conversation. In the Conversation phase, the CRS client exchanges information with the CRS server. The CRS foils the censor's attempts to block the user-to-CRS link, to tamper with the information being carried, or to identify distinguishers that can be subsequently used for blocking. The CRS client connects to the CRS server (typically outside the censor's sphere of influence), which connects to the publisher. The CRS server acts like a proxy between the user and the publisher, relaying traffic back and forth over

an unblockable channel. The communication between the CRS client and the CRS server can optionally be supported by CRS facilitators. The publisher is the entity which the CRS client ultimately wants to access; for example, 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). Alternatively, the publisher could also be the stepping stone to another blocked system. For example, Tor is often blocked by blacklisting its entry points, the bridge nodes. To get around the block, users employ a CRS to connect to the bridge nodes, subsequently bootstrapping into the Tor network. In this example, the bridge node is the publisher from the perspective of the CRS client.

In general, CRSes treat the dissemination link and the user-to-CRS link separately from the CRS-to-publisher link as these lend themselves to different design, implementation, and software distribution practices. Most CRSes provide circumvention on the user-to-CRS link because of its flashpoint status in the censorship arms race; censorship on this link is less intrusive and more convenient for the censor as most of the communication infrastructure is within its sphere of influence.

In contrast, the CRS-to-publisher link is usually outside the censor's sphere of influence; offering only simple access to the publisher (e.g. a HTTP or SOCKS proxy), without any security properties. Alternatively, the CRS-to-publisher link may connect to an anonymity system such as Tor [47] to offer privacy in addition to blocking-resistant access to information.

Example. To tie things together, we provide the example of a CRS, ScrambleSuit [48]; illustrating the phases of Communication Establishment and Conversation, and the interaction between various components. In the Communication Establishment phase, the CRS client retrieves a short-lived ticket from the CRS over a low-bandwidth out-of-band channel (e.g. a USB memory stick sent by regular post). The out-of-band channel is used only once when the CRS client joins ScrambleSuit for the first time. The CRS client redeems the ticket from the dissemination server for mutual authentication. After successful authentication, the dissemination server gives the client a ticket for the next connection to relieve the CRS client from having to retrieve tickets for subsequent connections. The CRS client includes this ticket in its TLS connection request to the CRS server. The CRS server only responds to a connection request if it contains a valid ticket, to prevent being fingerprinted by a probing censor. After the ticket presented by the CRS client is validated by the CRS server, the handshake is completed and the Conversation phase proceeds. In the Conversation phase, the application data can be exchanged over the encrypted TLS connection. Additionally, the CRS client and CRS server transform their traffic into random-looking bytes to thwart content-based fingerprinting, and remove flow fingerprints by randomizing packet lengths and flow timing.

2.3 Systematization Methodology

We now describe our methodology to evaluate censorship resistance systems. We conducted a comprehensive survey of CRSes—totalling 73 systems—both deployed and proposed in academic publications until February 2016. We selected academic publications that appeared in well-respected venues in security research. We selected deployed tools from references of the surveyed academic papers and results of searching on Google Scholar using relevant keywords. The full list of surveyed CRSes can be found in Appendix A.

Our CRS evaluation spans four dimensions—security, privacy, performance, and deployability— which are further broken down into specific properties. Instead of assessing individual systems, we evaluate *schemes* representing recurring themes and underlying concepts in the CRS landscape, to visualize strengths, limitations, and opportunities. Next we evaluate representative CRSes corresponding to each CRS scheme along the four dimensions of security, privacy, performance, and deployability. Note that a CRS may well employ multiple schemes in a layered design, and some properties of the representative CRS might map to multiple schemes; we flag such cases where applicable.

We apply the evaluation methodology sketched above to each of the two phases of CRS functionality, Communication Establishment and Conversation. Most properties are common across both Communication Establishment and Conversation (we state where this is not the case), and have binary values (*has*, *does not have*) with some exceptions that also have an intermediate value (*partially has*). On the whole, this methodology may be described as a semi-structured approach to CRS evaluation.

The evaluation process was conducted by six domain experts (including the author), and involved the following steps:

- **Survey and categorization.** Two groups of two experts studied all the 73 CRSes identified, generating two lists of CRS schemes and a representative system for each of these schemes. The two lists were consolidated into one and further refined through discussion among the four contributing experts.

- **Developing evaluation framework.** Two experts developed the framework to evaluate CRSes, which was iteratively refined through discussion with the other two experts.

- **Evaluation.** Once the evaluation framework, CRS strategies, and representative CRSes had been identified, one expert evaluated all the representative systems: she divided all the evaluated CRSes into three sets (each set contained CRSes corresponding to a diverse mix of CRS schemes) and invited the other three experts to independently verify the evaluation of these systems. Differences were resolved through iterative discussions and incorporated in the evaluation.

- **Verification.** To ensure inter-rater reliability, the other two (so far uninvolved) experts verified soundness of the evaluation process and the scores assigned to CRSes.

Usability is another important dimension of CRS evaluation alongside security, privacy, performance, and deployability. In this chapter, we restrict our scope to the first four of these and defer usability to future work; evaluation of CRS usability is not well understood, and forms an emerging research area.

2.3.1 Security Properties

CRSes incorporate a number of security properties: to prevent or slow the censor from learning high-quality distinguishers, to make it hard for the censor to distinguish CRS traffic from allowed network flows, and to ensure CRS availability even after it has been detected (e.g. by increasing the cost incurred by the censor to process network traffic). We define CRS security properties as follows:

Unobservability. The censor cannot detect prohibited communication or the use of CRS itself based on content or flow-based signatures, or destinations associated with the CRS or those known to serve prohibited content. Three common techniques that help achieve unobservability are as follows:

- **Content Obfuscation.** Communication exchanges between the CRS client and other CRS components do not contain unique static strings or string patterns that the censor can associate with prohibited material or the CRS itself.
- **Flow Obfuscation.** Communication exchanges between the CRS client and other CRS components do not contain unique stochastic patterns (e.g. packet sizes and timing) that the censor can associate with prohibited material or the CRS itself.
- **Destination Obfuscation.** The identities and network locations (e.g. IP addresses and domain names) of the dissemination server, CRS server, and key facilitators are hidden.

Unblockability. The censor, even after having identified prohibited content or use of the CRS, is either unable or unwilling to block communications (often because of the associated collateral damage). Two common techniques of unblockability are as follows:

- **Outside Censor's Influence.** This technique involves placing critical CRS components beyond the censor's sphere of influence (and sometimes sphere of visibility) such that the censor is unable to launch an effective attack even after identifying the CRS components.
- **Increase Censor's Cost to Block.** The censor's blocking decision depends on the accuracy of distinguishers and the blocking mechanisms employed. In particular, the censor's policy must make consideration for the acceptable false positive rate as these have political and economic ramifications [49]. Mechanisms under this category

obfuscate CRS credentials or infrastructure in such a way that blocking these incurs unacceptable collateral damage.

Availability. Does the CRS incorporate mitigation against DoS attacks on its components (either directly or by leveraging a DoS-resistant participant)? In general, there is a lack of robust techniques to completely neutralize such attacks; a common mitigation strategy is to over-provision bandwidth and IP addresses.

Communication Integrity. Information and the mechanisms to enable its retrieval are robust in the presence of a censor that tampers with information, actively manipulates channel properties (e.g. injecting, modifying, or dropping packets), or changes routing assumptions.

- **Message Integrity.** Does the CRS verify integrity of data in the face of an active censor that can tamper with data while in transit or stored on the publisher?
- **Server/Publisher Authentication.** Does the CRS incorporate authentication of the dissemination and CRS servers (Communication Establishment phase), and the publisher (Conversation phase)?
- **Packet Loss Support.** Is the CRS resilient against packet drops induced by the censor? (While packet drops can affect Communication Establishment as well as Conversation, we consider it only in the latter case where the threat is more pressing because of the volume of the information exchanged.)
- **Out-of-Order Packets Support.** Can the CRS handle packets that arrive out-of-order in the face of a censor that can inject random delays into traffic? (Applies to Conversation phase only.)

2.3.2 Privacy Properties

CRSes incorporate privacy properties to provide coercion resistance to CRS components so that the censor's threats of force are of limited impact, such as when the CRS component is outside the censor's sphere of influence, or the system is designed in a way that makes it technically impossible to comply with the censor's demands. We define CRS privacy properties as follows.

User Anonymity. Can the CRS client anonymously retrieve information from the dissemination server (Communication Establishment) and the publisher (Conversation)? The threat is from a censor that can enumerate (and subsequently coerce) users by observing connections to the dissemination server and the publisher.

Server/Publisher Anonymity. Can the dissemination server (Communication Establishment) and the publisher (Conversation) disseminate information to users without revealing their identity? The threat is from a censor that can identify (and subsequently coerce) servers and publishers of information by observing where CRS clients connect to (e.g. by masquerading as a CRS client) or where prohibited information is fetched from (e.g. through passive analysis of data on the wire).

User Deniability. The censor cannot reasonably confirm if the user intentionally accessed prohibited information or used the CRS, therefore is unable to implicate users because of insufficient evidence. A common approach to enforce user deniability is by transforming traffic to an obfuscated form when it leaves user machine (e.g. through encryption or steganography).

Server/Publisher Deniability. The censor cannot reasonably confirm if the dissemination server or the publisher intentionally served prohibited information or participated in the CRS, therefore is unable to implicate the two because of insufficient evidence. This can be enforced by obfuscating responses of the dissemination server and the publisher (e.g. through encryption or steganography).

Participant Deniability. Parties that support CRS operations should be able to deny intentional participation to prevent the censor from blocking or punishing them.

2.3.3 Performance Properties

CRSes have certain performance characteristics which in some cases are the side effect of the CRS scheme employed and cannot be tuned; in other cases, performance can be improved by making appropriate design decisions. We define CRS performance properties as follows.

Latency. For Communication Establishment, latency corresponds to the time lag between the CRS client initiating Communication Establishment and when it is ready to start Conversation (compared to the baseline latency when directly connecting to the CRS server). In the case of Conversation, latency is the delay introduced by the CRS compared to the baseline approach of directly downloading a document over a standard protocol like HTTP. A number of factors can contribute to CRS communication latency; for example, artificial inter-arrival times between packets, packet padding, and the additional CRS protocol header.

Goodput. This refers to the useful bandwidth available for the information originally requested by the CRS client; that is, the total bandwidth minus the CRS's operational overhead. A high-goodput CRS implies that the CRS achieves high-throughput while

incurring low operational overhead, resulting in high bandwidth available for the CRS client to retrieve information. A low-goodput CRS means that either the CRS is inherently low-throughput; or that even though the CRS is high-throughput, its high overhead leaves little bandwidth for the information requested by the CRS client. (Applies to Conversation phase only.)

Stability. Do the performance characteristics of the CRS (e.g. communication latency and goodput) remain consistent? For each scheme, we identify factors (e.g. reliance on external conditions beyond CRS control) that cause performance to fluctuate, and to what degree.

Scalability. How well does the CRS scale when there is an increase in the number of CRS clients that want to communicate with the dissemination server or access information using the CRS server? For each scheme, we identify factors that affect scalability, and to what degree.

Computational Overhead. The degree of additional computational resources incurred by the dissemination server (compared to the baseline approach of directly connecting to the CRS server) and the CRS server (compared to the baseline approach of directly downloading a Web page) as a result of using the CRS. In Communication Establishment, computational overhead is typically proportional to the cryptographic overhead introduced by the authentication scheme used by the dissemination server.

Storage Overhead. The degree of additional storage required by the dissemination server and the CRS server because of the CRS compared to the baselines described for computational overhead.

2.3.4 Deployability Properties

The utility of a CRS depends not only on its security, privacy, and performance properties, but also on how amenable it is to be deployed and used in the real world. We define CRS deployability properties as follows.

Synchronicity. Does the CRS require all the components relevant to the Communication Establishment and Conversation phase to be online at the same time?

Network Agnostic. Do the Communication Establishment and Conversation phases make specific network layer assumptions (e.g. specific routing requirements or assumptions about packet fragmentation and TTL values)?

Coverage. This refers to the degree of Internet access (expressed as the fraction of the Web, Internet services, and protocols) enabled by the CRS. A CRS is rated as high-coverage if it can be used to access any content, medium-coverage if there are restrictions on the type of content or length, and low-coverage if both the content type and length are restricted. (Applies to Conversation phase only.)

Participation. A large number of CRSes depend on cooperation from participants—intermediate actors that facilitate CRS activities such as proxies (Section 2.2). We define the following metrics to assess the quality and flexibility of cooperation required by CRSes from participants.

- **Quantitative Incentivization.** Is the incentive structure to encourage participants to help the Conversation based on tangible rewards (whether monetary or in kind) instead of qualitative incentives such as goodwill and public relations?

- **Distributed Participation.** This represents the degree to which participation is diffused among cooperating entities: a low value corresponds to a single organized entity (e.g. a corporation, a company, or an institution), a medium value represents multiple organized bodies, and a high value indicates individual volunteers participating in personal capacity.

- **Voluntary Participation.** Is the participant aware that the CRS has employed it for censorship resistance?

- **Conditional Participation.** Is the participation premised on specific conditions such as popularity, reputation, or location?

- **Deterministic Cost.** Does the CRS provide estimated cost to participants, or alternatively allow them to control the degree of participation in the CRS?

- **Security Delegation.** Are the security properties offered by the CRS compromised when there are subverted participants?

- **Privacy Delegation.** Are the privacy properties offered by the CRS compromised when there are subverted participants?

2.4 Communication Establishment

We describe schemes employed by CRSes to enable the CRS client to obtain CRS credentials; while preventing the censor from detecting and blocking Communication Establishment, or harvesting and fingerprinting CRS credentials.

2.4.1 High Churn Access

This scheme relies on the selection of CRS credentials that change regularly such that these cannot be preemptively blocked by the censor. There is a window of opportunity, between employing new CRS credentials and until the censor discovers them, where misclassification can occur.

2.4.2 Rate-Limited Access

To prevent the censor from harvesting CRS credentials by masquerading as a legitimate CRS participant, the CRS may limit the rate at which CRS credentials can be queried. Rate-limiting is usually done by requiring proof-of-work from participants, partitioning the value space over time slices and based on CRS participant attribute(s), or employing reputation-based checks.

Proof-of-Work. To prevent the censor from harvesting CRS information at scale, some CRSes employ proof of work approaches such as CAPTCHAs and puzzles; the assumption is that such puzzles are too expensive for wide-scale harvesting by a resource-bounded censor.

Time Partitioning. In this scheme, the CRS partitions CRS credentials over time slices (e.g. by creating a large pool of unpredictable values, and employing a subset of these values for short time periods). As a result, the censor needs to continuously allocate resources to stay up-to-date with CRS values.

Keyspace Partitioning. In this scheme, CRS information is partitioned over attribute(s) specific to users (e.g. client IP address). As a result, each user only learns a restricted set of CRS credentials from the entire value space.

2.4.3 Active Probing Resistance Schemes

A censor may probe suspected dissemination and CRS servers to confirm if these participate in the CRS. To mitigate this threat, CRSes introduce a sequence of steps during the Communication Establishment phase; these steps are feasible for a single CRS user to follow, but hard for a censor to perform at scale. The servers can obfuscate their association with the CRS from unauthenticated users by pretending to be offline (*obfuscating aliveness*), or by providing an innocuous response or no response at all (*obfuscating service*).

Obfuscating Aliveness. A dissemination server may not respond to connection requests from a CRS client until the client completes an expected sequence of steps (e.g. embedding a secret token in the request, or encoding a valid request in a series of packets sent to predefined ports in a specific order).

Table 2.1: Evaluation of censorship resistance schemes related to the Communication Establishment phase of CRS functionality. Notation for binary values: ✓ has property, ✗ does not have property. Notation for non-binary values: ● has property, ◒ partially has property, ○ does not have property. – means the property does not apply to the given scheme. The last row provides a breakdown of deployment status of *all* the systems surveyed for the given scheme; a full list of the corresponding citations is provided in Appendix A.

	High Churn Access	Rate Limited			Active Probing		Trust-Based Access
		Proof of Life/Work	Time Partitioning	Keyspace Partitioning	Obfuscating Aliveness	Obfuscating Service	
System	Flashproxy [50]	Defiance [51]	Tor Bridges [52]	Keyspace-Hopping [53]	SilentKnock [54]	ScrambleSuit [48]	Proximax [55]
Security							
Unobservability							
Content Obfuscation	✗	✓	✓	✓	✓	✓	✗
Flow Obfuscation	✗	✗	✗	✗	✗	✓	✗
Destination Obfuscation	✓	✓	✓	✓	✓	✓	✓
Unblockability							
Outside Censor Influence	✓	✓	✓	✓	✓	✓	✗
Increase Censor Cost to Block	✓	✓	✓	✓	✓	✓	✓
Availability	✓	✓	✓	✓	✓	✓	✓
Communication Integrity							
Server Authentication	✗	✓	✓	✗	✗	✗	✗
Message Integrity	✗	✓	✓	✗	✓	✓	✗
Privacy							
User Anonymity	✗	✗	✗	✗	✗	✗	✗
Publisher/Server Anonymity	✗	✗	✗	✗	✗	✗	✗
User Deniability	✗	✓	✓	✓	✓	✓	✗
Publisher/Server Deniability	✗	✓	✓	✓	✓	✓	✗
Participant Deniability	✗	✓	✓	✓	–	–	✗
Performance							
Low Latency	●	○	●	●	◒	◒	●
Stability	○	○	●	◒	●	●	○
Scalability	○	○	○	○	●	◒	○
Low Computation	●	○	●	●	◒	◒	●
Low Storage Overhead	●	◒	●	●	●	●	●

Continued on next page

Table 2.1 (continued from previous page)

	High Churn Access	Rate Limited			Active Probing		Trust-Based Access
		Proof of Life/Work	Time Partitioning	Keyspace Partitioning	Obfuscating Aliveness	Obfuscating Service	
System	Flashproxy [50]	Defiance [51]	Tor Bridges [52]	Keyspace-Hopping [53]	SilentKnock [54]	ScrambleSuit [48]	Proximax [55]
Deployability							
Synchronicity	◒	◒	●	●	●	●	◒
Network Agnosticism	✓	✓	✓	✓	✓	✓	✓
Participation							
Quantitative Incentivization	✗	✗	✗	✗	–	–	✗
Distributed Participation	●	●	●	●	–	–	●
Voluntary Participation	✓	✓	✓	✓	–	–	✓
Conditional Participation	✗	✗	✓	✗	–	–	✓
Deterministic Cost	✓	✗	✓	✗	–	–	–
Security Delegation	✓	✓	✓	✓	–	–	✓
Privacy Delegation	✓	✓	✓	✓	–	–	✓
Status							
Academic & Tool available	2	0	0	0	2	1	0
Academic paper only	0	2	0	1	0	0	1
Tool only	0	0	1	0	0	0	1

Obfuscating Service. In this scheme, the dissemination server responds to connection requests from a CRS client (thus revealing its aliveness), but refuses to speak the CRS protocol until the CRS client completes a predefined sequence of steps.

2.4.4 Trust-Based Access

The server may associate an element of trust with CRS users based on their previous behaviour (or derived from their social network graph), responding only to requests from CRS clients with satisfactory reputation.

2.4.5 Discussion

In Table 2.1, we present our evaluation of CRSes related to the Communication Establishment phase of CRS functionality. The rows correspond to the security, privacy, performance, and deployability properties described in Section 2.3. The columns represent our evaluation of CRS schemes in the context of representative CRSes. The last row of the table provides a breakdown of the deployment status of all the surveyed CRSes for each scheme (a complete list of citations can be found in Table A.1 in Appendix A). A majority of CRSes address threats in Conversation: we found 62 systems concerning Conversation, and only 11 systems relevant to Communication Establishment. Tschantz *et al.* note that the research emphasis on Conversation is orthogonal to real censorship attacks which tend to concentrate on Communication Establishment [56]. We now present the results from our evaluation, highlighting common trends and discussing the strengths and limitations of the CRS schemes. Note that the effectiveness of a given CRS depends on the context in which it is employed; the goal of our evaluation is to characterize the suitability of CRS schemes to different use cases and censor capabilities.

We observe that the schemes in which the server responds after validating users (active probing resistance, proof of work) typically offer *content obfuscation*; for example, by ensuring that messages containing credentials such as puzzles and tokens do not have content-based signatures. A naïve active probing resistance system that includes a fixed-length token in the request is vulnerable to *flow fingerprinting*; the censor can detect connections that always begin with sending a fixed number of bytes. Such length-based signatures can be removed by using pseudo-random padding [48]. With respect to performance, the user validation step in these schemes comes at the cost of increased *time to join the system*, higher *computational needs*, and possibly additional *storage requirements* if the server stores identifiers for static matching instead of verifying identifiers at runtime. As disparate interactions are required between the CRS client and different CRS servers and participants, these can be at least partially conducted *asynchronously* (usually within some time window). Schemes that obfuscate service need more consideration with respect to *load balancing*; because such schemes perform validation at the application layer, for each request a TCP connection is still established.

Most trust-based and high churn access schemes do not *obfuscate content*; this functionality needs to be built on top of these schemes. The same is true of *flow obfuscation*: Communication establishment involves just a few brief interactions that are insufficient for statistical classifiers (especially those based on inter-packet arrival times) which require a large volume of data to produce high quality results. Schemes that involve straight-forward access to CRS servers with some prudence on the part of the server to hand out values with discretion (e.g. high churn access, time partitioning, keyspace hopping, proxy, and trust-based schemes) tend to have low *latency*. In general, there is a single interaction between the CRS client and the CRS server (either directly or through intermediate forwarding participants); consequently, the client, the server, and all participants need to be *online at the same time*.

A particular consideration for high churn access systems is that new values must be constantly available. However, if these values are opportunistically derived from volunteer participants, the system is neither *stable* nor *scalable*: It is not clear how long a value will remain available and how many values will be available as the number of users increases. A possible mitigation is to have a notion of participant quality (e.g. in terms of available bandwidth and uptime) so that requests are more intelligently distributed under increased demand (*conditional participation*). We note that conditional participation enables the CRS to adequately plan issues concerning system scalability and stability. In a broader context, we observe that most participant-based schemes recruit individual entities who volunteer to help based on *qualitative incentivization* (e.g. goodwill): all such schemes fail to offer privacy and security if participants are partially or completely under the censor's control. As a result, any participant can potentially compromise security and privacy properties of the CRS; in such cases, decentralized systems fare better in terms of damage control.

A shortcoming of rate-limited access and active probing resistance schemes is that the censor can sometimes deploy more resources than anticipated to harvest the CRS value space, effectively neutralizing the benefits of these schemes (the Sybil attack). For example, to neutralize proof of work schemes, the censor can invest in computational power to solve multiple puzzles in parallel [57].

In high churn access, the CRS value space harvested by the censor is quickly outdated; consequently, distinguishers harvested by the censor are unstable and inconsistent, requiring frequent updates. A challenge for high churn access schemes is how to manage the value space so that new values are constantly available, especially when the value space is contingent on volunteers whom the CRS cannot directly control. Trust-based schemes are vulnerable to malicious participants: To gain access to a CRS employing trust-based access, the censor can impersonate a credible user (e.g. by stealing credentials of an existing user or earning good reputation over a period of time, both of which do not scale well). The threat from malicious participants is tackled by tracking CRS user reputation even after initial authorization; malicious behaviour can lead to ejection. However, it can potentially

take only a few subverted CRS users to deplete and block the entire value space: this can be mitigated by rate-limiting the information available per CRS user.

Most schemes do not incorporate *authentication* of dissemination and CRS servers; as a result, the censor can masquerade as a server and enumerate and coerce users. To mitigate this threat, the CRS can maintain a centralized authority which the CRS client can query independently to establish authenticity of a server (e.g. Tor directory services [58]). Alternatively, the censor can masquerade as a server and disrupt CRS accessibility by distributing bogus CRS credentials (especially relevant if the CRS design includes volunteer servers which cannot be authenticated). As a mitigation, the CRS can digitally sign the information distributed by servers so that the CRS users can verify the *integrity of information* obtained from untrusted servers (e.g. Defiance NET payloads). For the same reason, most probing-resistant schemes offer message integrity so that the censor cannot tamper with the information that validates the CRS user to the server. On the other hand, we note that most schemes (high churn access, keyspace hopping, trust-based access) that manage control to unauthenticated proxies do not offer message integrity, expecting this to be handled by another application on top of it.

With respect to *privacy*, none of the schemes offer *user anonymity*: if the censor successfully fingerprints credentials of CRS servers, it can identify users that connect to these. However, it may prove difficult to implicate users at this point because there has only been an attempt to use the CRS—the CRS has not been used to access blocked content. The same applies to *server anonymity*: the censor can masquerade as a legitimate CRS user and identify the servers that the CRS client contacts. Most schemes offer *user and server deniability* by using encryption or steganography which to some extent compensates for the lack of anonymity. Trust-based schemes are an exception in setting up stringent criteria for CRS participation, but such schemes do not adequately address user deniability if CRS members have been subverted. Schemes that rely on volunteer participants have an obvious incentive to offer *participant deniability*. Some schemes waive this property, assuming that the participant is outside the censor's sphere of influence and therefore immune to coercion attempts.

2.5 Conversation

In this section, we describe schemes employed by CRSes to evade detection and blocking in the Conversation phase. We note that the identified schemes enable unobservable and unblockable access to information (access-centric) and may incorporate additional measures to store information for increased availability and coercion resistance (publication-centric). For clarity, we observe this distinction in our discussion; however, it is possible for a CRS to employ schemes from both groups.

2.5.1 Access-Centric Schemes

Schemes in this category protect access to information over the user-to-CRS link (Figure 2.4) by safeguarding security and privacy of relevant CRS components, and by protecting information in transit from corruption.

Mimicry. This scheme transforms traffic to look like whitelisted communication such that the transformed traffic resembles the syntax or content of an allowed protocol. Mimicry can be of known protocols or randomness, and of content or flows.

Content Mimicry evades censorship by imitating the syntax of an innocuous protocol (e.g. HTTP) or content (e.g. HTML). Typically content mimicry is based on widely-deployed protocols or popular content, with the goal of complicating the censor's task by: *(i)* increasing the censor's work load because of the large volume of traffic that needs to be inspected, and *(ii)* increasing the collateral damage associated with wholesale protocol blocking. Another approach is to make traffic content look like an unknown protocol, either by imitating randomness or by arbitrarily deviating from a known blocked protocol. This idea is motivated by the general assumption that the censor only implements blacklisting of known 'bad' protocols, and is unwilling to incur the high collateral damage associated with whitelisting.

Flow Mimicry is similar to content mimicry except that the CRS imitates flow-level characteristics (e.g. packet length and inter-arrival timings) of unblocked protocols, or randomizes flow-level characteristics to remove statistical fingerprints.

Tunnelling. In contrast to mimicry where the CRS pretends to be an unblocked protocol, managing to only partially mimic the target protocol; in this scheme, the CRS traffic is tunnelled through an unblocked application enabling nearly flawless mimicry.

Covert Channel. In this scheme, CRS communication is hidden in a cover medium, creating a covert channel within the cover that transmits CRS traffic in novel ways (e.g. hiding HTTP requests within a cover image). As a result, CRS traffic is not only hard to detect, but also enables deniability for both users and publishers (e.g. through steganographic techniques).

Traffic Manipulation. This scheme exploits the limitations of the censor's traffic analysis model, shaping CRS traffic such that the censor is unable to fingerprint it. Effectively, this scheme renders even cleartext traffic unobservable; this property is particularly useful for CRS users in countries that prohibit encrypted traffic.

Destination Obfuscation. To prevent the censor from observing and blocking the key destinations that the CRS client directly connects to, the client can instead relay CRS traffic through one or more intermediate nodes to obfuscate the original destinations.

Table 2.2: Evaluation of censorship resistance schemes related to the Conversation phase of CRS functionality. Notation for binary values: ✓ has property, ✗ does not have property. Notation for non-binary values: ● has property, ◒ partially has property, ○ does not have property. – means the property does not apply to the given scheme. The last row provides a breakdown of deployment status of *all* the systems surveyed for the given scheme; a full list of the corresponding citations is provided in Appendix A.

	Access-Centric Schemes						Publication-Centric Schemes	
	Content/Flow Obfuscation				Destination Obfuscation		Content Redundancy	Distributed Storage
	Mimicry	Tunnelling	Covert Channel	Traffic Manip.	Proxy	Decoy Routing		
System	SkypeMorph [59]	Freewave [60]	Collage [61]	Khattak *et al.* [62]	Tor [47]	Cirripede [63]	Freenet [64]	Tangler [65]
Security								
Unobservability								
Content Obfuscation	✓	✓	✓	✓	✓	✓	✗	✗
Flow Obfuscation	✓	✗	–	✗	✗	✓	✗	✗
Destination Obfuscation	✗	✓	✓	✗	✓	✓	✓	✓
Unblockability								
Outside Censor Influence	✓	✓	✓	✓	✓	✓	✗	✓
Increase Censor Cost to Block	✓	✓	✓	✓	✓	✓	✓	✓
Availability	✗	✗	✓	✗	✗	✗	✓	✓
Communication Integrity								
Publisher Authentication	✓	✗	✓	✗	✗	✗	✓	✓
Message Integrity	✓	✗	✗	✗	✓	✓	✓	✓
Packet Drop Resistance	✗	✗	–	✗	✗	✓	–	–
Out-of-Order Resistance	✓	✓	–	✗	✓	✓	–	–
Privacy								
User Anonymity	✗	✓	✓	✗	✓	✓	✓	✓
Publisher Anonymity	✗	✓	✓	✗	✓	✓	✓	✓
User Deniability	✓	✓	✓	✗	✓	✓	✗	✓
Publisher Deniability	✓	✗	✓	✗	✗	✗	✗	✓
Participant Deniability	✓	✓	✓	–	✓	✓	✗	✓

Continued on next page

Table 2.2 (continued from previous page)

	Access-Centric Schemes						Publication-Centric Schemes	
	Content/Flow Obfuscation				Destination Obfuscation		Content Redundancy	Distributed Storage
	Mimicry	Tunnelling	Covert Channel	Traffic Manip.	Proxy	Decoy Routing		
System	SkypeMorph [59]	Freewave [60]	Collage [61]	Khattak *et al.* [62]	Tor [47]	Cirripede [63]	Freenet [64]	Tangler [65]
Performance								
Low Latency	◒	◒	○	◒	◒	●	●	◒
High Goodput	◒	◒	○	●	●	●	–	–
Stability	●	●	○	◒	●	●	◒	●
Scalability	◒	●	○	○	◒	◒	◒	◒
Low Computation	◒	◒	◒	●	●	◒	●	◒
Low Storage Overhead	●	●	○	●	●	●	●	●
Deployability								
Synchronicity	✓	✓	✗	✓	✓	✓	✗	✗
Network Agnosticism	✓	✓	–	✗	✓	✗	✓	✓
Coverage	●	●	○	●	●	●	◒	◒
Participation								
Quantitative Incentivization	✗	✗	✗	–	✗	✗	✗	✓
Distributed Participation	○	◒	◒	–	●	◒	●	●
Voluntary Participation	✗	✗	✗	–	✓	✓	✓	✓
Conditional Participation	✓	✓	✓	–	✓	✗	✓	✓
Deterministic Cost	✗	✗	✗	–	✓	✗	✗	✓
Security Delegation	✓	✓	✓	–	✓	✓	✓	✓
Privacy Delegation	✓	✓	✓	–	✓	✓	✓	✓
Status								
Academic & Tool available	2	0	1	2	4	0	1	0
Academic paper only	8	7	7	2	4	5	4	1
Tool only	5	2	0	1	10	0	0	0

The CRS may employ a *proxy* as CRS facilitator to relay traffic between the CRS client and the CRS server. Usually, the proxies change frequently to avoid getting blocked. To further improve privacy, the CRS traffic can be relayed through multiple proxies.

In *decoy routing*, clients covertly signal a cooperating intermediate router to deflect their traffic purportedly en route to an unblocked destination (dummy destination to evade the censor) to a blocked one (the intended destination). The deflecting routers must be located on the forward network path from the client to the unblocked destination.

2.5.2 Publication-Centric Schemes

Schemes in this category protect information over the CRS-to-publisher link (Figure 2.4). Systems in this scheme allow publishers to push information to the CRS which is stored among multiple CRS servers, and served to CRS users upon request. Consequently, the main goal of publication-centric schemes is to ensure availability and integrity of information, while offering deniability to CRS servers, and preferably also to CRS users and publishers. It is common for these schemes to refer to published information as *documents*, hence we use the terms interchangeably.

Content Redundancy. This scheme stores content redundantly on a large number of CRS servers ideally placed in different jurisdictions to make it hard for the censor to remove prohibited content from all the servers.

Distributed Content Storage. In this scheme, the content is broken into smaller chunks which are distributed among multiple CRS servers so that none of the servers has the full document. To reconstruct the document, the corresponding chunks are retrieved from the CRS servers where these are stored.

2.5.3 Discussion

In Table 2.2, we present our evaluation of censorship resistance schemes related to the Conversation phase of CRS functionality. The rows correspond to the security, privacy, performance and deployability properties described in Section 2.3; and the columns represent evaluation of the defined CRS schemes along these properties. Our evaluation is based on representative CRSes corresponding to each CRS scheme. The last row of the table shows a breakdown of the deployment status of all the CRSes for each CRS scheme (a complete list of citations for these systems can be found in Table A.2 in Appendix A). We surveyed 62 systems, of which 28 have end user tools available; most of the end user tools represent mimicry and proxy-based schemes. We now discuss common trends as well as strengths and limitations of CRS schemes based on our evaluation. Our goal is to characterize the suitability of the CRS schemes to different use cases and censor capabilities.

We observe that nearly all access-centric schemes offer *content obfuscation*, but only mimicry schemes *obfuscate traffic flows*. Some mimicry schemes morph traffic to resemble the content or flow-level characteristics of a cover protocol. This approach is inherently imperfect as cover protocols are generally complex, with disparities stemming from incomplete or incorrect cover protocol imitation (e.g. failure to handle errors in a consistent manner). The censor can leverage such disparities to identify CRS traffic through active manipulation [66] [67] [68]. Some schemes morph traffic such that its content and flow-based features look random, making it hard for the censor to match the traffic to any known protocol. However, if the censor only allows whitelisted protocols then it can flag random looking traffic as anomalous and block it. Mimicry-based systems offer user deniability, but destination obfuscation and privacy properties are generally lacking.

Tunnelling ameliorates the limitations of mimicking cover protocols to some degree (such as *destination obfuscation*, and resistance to *dropped packets*); however, the censor may be able to take advantage of inconsistencies in channel usage or content. CRS traffic may still have flow-based features that distinguish it from the cover protocol [66]. Furthermore, the CRS may rely on channel characteristics in a different manner from the cover protocol; for example, if the cover protocol is more robust to network degradation, the censor can manipulate the network to disrupt CRS traffic without seriously affecting legitimate cover protocol traffic [69]. Tunnelling systems typically leverage popular third-party platforms to increase *collateral damage* associated with blocking the CRS. As such platforms are provisioned for Internet-scale performance, tunnelling-based systems also inherit high *availability* and *scalability*. Both mimicry and tunnelling schemes incur additional protocol overhead, resulting in decreased *goodput* and additional *latency* in extracting CRS traffic from the cover (e.g. demodulating voice data received over a cover VoIP application to recover tunnelled information). Despite its strengths, tunnelling schemes have mainly received attention in academic literature, with only two tools available (Bit-Smuggler [70] and YourFreedom [71]).

In our survey, proxy-based schemes have the highest number of end user tools available. Both proxy-based schemes and decoy routing relay traffic through intermediate participants. In the former case, traffic can be redirected through multiple proxies, which can lead to an increase in *latency*. A crucial component of the decoy routing strategy is that by building circumvention into the Internet infrastructure, the need for communication establishment is obviated. The CRS client includes the credentials needed to join the CRS in a covert channel inside its request for an unblocked overt destination; a decoy router intercepts the connection on path to the overt destination and deflects it to a proxy that facilitates communication with the blocked destination. Additionally, the use of TLS enables *destination obfuscation*, *content obfuscation*, good *performance*, and resistance to *active manipulation* (we find in our evaluation that most decoy routing systems are resilient against attacks involving dropped packets). The requirement for decoy routers to be deployed in cooperative ISPs is based on the assumption that the censor cannot

"route around" cooperating ISPs without significant *collateral damage*. However, if this assumption proves to be invalid, the censor can avoid or otherwise blackhole such a route and nullify this scheme [72]. We note that none of the decoy routing systems have been deployed; the requirement to be supported by real-world ISPs poses a significant deployment challenge. Moreover, the *scalability* of these systems is dependent on the number and location of the supporting ISPs; decoy routers are expected to be deployed widely enough to be on path to a large number of overt destinations.

Traffic manipulation schemes use low level network tricks to cause the censor to misinterpret traffic flows, *failing to detect blocked content*. Traffic manipulation schemes make certain assumptions about censorship apparatus; as a result, such schemes are vulnerable to attacks involving *out-of-order packets*. Another limitation is that these schemes fail to offer good *performance* and *scalability* if the censor's policy changes frequently, causing the CRS to be tuned and updated according to individual censorship policies.

Covert channel-based systems have been popular in academic literature, but there is only one such tool available (Infranet [73]). This scheme provides *unobservable communication* while maintaining a high degree of *privacy* for all CRS components. Like tunnelling-based systems, the robustness of the scheme depends on how closely the CRS traffic blends into content of the cover medium and conforms to semantics of the cover protocol. Some cover media are vulnerable to specific attacks. Timing-based covert channels are sensitive to *dropped and out of order packets*. Covert channels built into header values of network protocols (e.g. timestamp, initial sequence numbers, padding values, and flags) can attract attention for being anomalous, and the covert channel can be destroyed if the censor normalizes fields that are typically unused.

Attaining good *performance* is a challenge for covert channel schemes. CRS traffic has to be encoded inside cover traffic; consequently, the amount of information that the cover traffic can carry (*goodput*) is limited. The *coverage* of these schemes is typically restricted to static short messages. Some recent systems [74] [75] that utilize online games as the cover application manage to achieve lower *latency*; but the issue with low goodput remains, and additional processing to extract CRS traffic from the cover medium adds to latency. Some systems might have high *storage* requirements if the server maintains a collection of cover media to embed CRS traffic.

There exist six publication-centric systems (which appeared in early 2000s) and only two such tools are available. The emphasis has shifted to access-centric schemes, possibly because properties traditionally associated with secure publication are now offered by content delivery networks. Publication-centric schemes have the goal to increase *availability* of stored information while offering *deniability* to publishers and participants. To protect documents from removal, some CRSes replicate documents across multiple participants (usually these are individual volunteers and their *participation* is potentially *conditioned* on the bandwidth or storage that they are willing to offer); consequently, the censor

has to invest additional resources to remove a document especially if the participants are based in different geographic jurisdictions. Another approach to increase document availability is to create dependence between multiple documents by intertwining them with each other; removing a document results in the deletion of other intertwined documents as well, causing collateral damage. Although most publication-centric schemes provide *publisher authentication, document integrity and privacy properties*, these schemes suffer from a major limitation: the protocol messages are observable which makes it possible for the censor to fingerprint the CRS. By design, publication-centric schemes tend to be *asynchronous*, providing partial *coverage* (allowing access to static documents, generally with no restriction on size). The *stability* of these schemes depends on the time it takes to find the information requested.

2.6 Related Work

The literature contains a number of surveys, taxonomies, and reports that analyze censorship resistance, their varying goals, scopes, and technical depth. Our study extends previous work by capturing the breadth of the entire field of censorship resistance, while also providing sufficient technical details.

Elahi and Goldberg sketch the censor's attack model, and present a taxonomy of censorship resistance strategies for different types of censorship classified by the censor's resources, capabilities, limitations, and utility [76]. Our study extends this work by providing more technical depth and a rigorous systematization methodology.

Khattak *et al.* concentrate on pluggable transports [77], a framework to allow access-centric CRSes to flexibly plug into a larger system like Tor [78]. They represent functionality of a CRS that protects the link between the CRS client and the CRS server as a layered stack, identifying threats and mitigations relevant to each layer. Their model of pluggable transports mirrors our own in that they are also concerned with disruption of access to information; however, our scope comprises the entire CRS landscape that also includes publishers, CRS participants, and various elements of communication establishment.

Tschantz *et al.* conduct an extensive survey of evaluation criteria used by CRSes, and compare these to the behaviour of real censors as reported in field reports and popular bug tickets [56]. They find a significant disconnect between the threat models employed in theoretical evaluations and how censors operate in practice. While their enumeration of criteria used by CRSes has some overlap with the security, privacy, performance, and deployability properties that we use in our systematization, the study has different goals than ours: we evaluate underlying CRS approaches, whereas they exhaustively enumerate evaluation criteria employed by CRSes to identify trends and how these relate to real world censors.

The above work is most closely related to our study. We now discuss other studies that are generally relevant to ours. Köpsell and Hillig present a classification of blocking

techniques based on the communication layer involved (TCP/IP), the content of communication, and metadata of the communication (e.g. IP addresses of the participants, duration of the communication, and protocols involved) [79]. Leberknight *et al.* discuss the social, political, and technical aspects that underpin censorship, and their effectiveness in terms of scale, cost, and granularity [80]. Perng *et al.* classify circumvention systems based on the technical primitives and the principles that they build on [81]. Tschantz *et al.* argue that the evaluation of circumvention tools should be based on economic models of censorship [49]. Gardner presents a non-technical document on freedom-supporting technologies, describing their maintenance and funding ecosystem, user demographics, and the factors governing their development and usage [82].

2.7 Open Areas and Research Challenges

We provided a comparative evaluation of existing censorship resistance approaches to characterize their suitability to different censorship models (security and privacy properties), use cases (performance properties), and deployability scenarios. In this section, we discuss the open areas and challenges highlighted by this study.

2.7.1 Modular System Design

Our evaluation of various CRS schemes (Sections 2.4.5 and 2.5.3) suggests that it may be possible to increase the coverage of desirable security and privacy properties by combining complementary CRSes [83]. However, this is not straightforward because most CRSes have been designed to be stand-alone systems with tightly integrated functionality. Even designs that share a common base such as the Tor network and the various *pluggable transports* (CRS systems that can interface with Tor in a plug-and-play fashion [77]) suffer from this problem. Although the Tor community is actively trying to address this problem with the pluggable transport framework [84], the desired level of modularity and composability within pluggable transports has not yet been achieved [78] [85]. Jumpbox alleviates the problem at the network interface layer by providing a standard interface for encapsulating CRS traffic to look like regular Web traffic [86].

Recently, some tools have combined pluggable transports; however, this has been done through the sharing of source code, rather than in a black-box way. LibFTE is in use by Tor (in its fteproxy Pluggable Transport form) and a number of other projects [87]. Similarly, Meek [88] was originally developed for Tor; but it now also exists in a fork by Psiphon [89] with minor adaptations. Fog uses multiple proxies to chain pluggable transports in a black box fashion [90]. This approach is not suitable for practical deployment: not all combinations of pluggable transports make sense. The chain obfs3 [91] (flow fingerprinting resistance) followed by Flashproxy [50] (IP address filtering resistance) offers more comprehensive resistance; but the reverse, that is Flashproxy followed by obfs3,

breaks the former's network layer assumptions. Khattak *et al.* provide a coarse framework to build access-centric CRSes out of reusable components [78].

Another challenge is that CRS designs generally have to make a trade-off between security and performance. Combining CRSes to create a hybrid usually comes at a performance cost. For example, combining a CRS that tunnels traffic through a popular content provider with a CRS that redirects traffic through multiple proxies to provide anonymity may provide stronger security and privacy, but its performance will likely be poor because of the tunnelling protocol's network overhead and the multi-hop routing employed by the second CRS. There might be some cases where the hybrid's performance profile remains the same, such as the combination of a publication-centric CRS (e.g. Tangler [65]) with an access-centric scheme that uses a covert channel to store CRS content on popular platforms (e.g. Collage [61]). The CRS user can use the covert channel to send requests to the CRS server which are served in the usual way. Internally, the CRS can replicate information over multiple CRS servers for high availability, allowing the publishers to deniably post information to the CRS. This kind of hybrid is effective for use cases that prioritize security properties and can tolerate low performance.

2.7.2 Revisiting Common Assumptions

Nearly all CRSes assume that increasing the censor's cost to block and placing servers and facilitators outside the censor's sphere of influence leads to unblockability. However, as we note below, these assumptions do not always hold.

While increasing the collateral damage incurred by the censor is a common CRS design strategy, the effect of false negatives on the censor's behaviour is not well-understood. Filling this gap is an important next step in illuminating the dynamics of the censorship resistance game. There has recently been some work in this area: Tschantz *et al.* present a simple model of the costs of censorship to illustrate that CRSes should be evaluated based on economic models of censorship [49], and Elahi *et al.* apply game-theoretic analysis to censorship resistance and investigate the effect of information leakage, collateral damage, and accuracy of the censorship apparatus on the censor's behaviour [83].

Another common CRS strategy is to resist fingerprinting attacks by mitigating perceived low-cost distinguishers. However, the capabilities and willingness of the censor to engage in sophisticated traffic analysis is unclear, and the cost of detecting complex high-cost distinguishers is not well-understood. While it is difficult to find out the true capabilities of a censor, it is still useful to assess the cost to the censor in employing low- and high-quality distinguishers. Existing literature includes analysis of the censor's threat models, but such studies usually have a narrow focus and it is not clear how these relate to real censors. It has been reported that the threat models employed by most existing CRSes are disconnected from how real censors operate; for example, real censors tend to avoid packet dropping attacks due to the large collateral damage [56]. For the same reason, mimicry and tunnelling-based schemes continue to be effective in practice despite

theoretical attacks that demonstrate their vulnerability to active manipulation [66] [67] (Section 2.5). Another CRS design aspect that suffers from a lack of robust evaluation is resistance against fingerprinting of flow properties and content (typically by shaping traffic using different protocols in a 'correct' manner). Such CRSes should be validated against labelled datasets of different types of network traffic (ideally maintained in a community repository). Adversary Lab [92] has done some preliminary work on developing a standard environment to evaluate CRSes that resist flow fingerprinting by subjecting them to a range of adversaries. There is a need for Internet-scale studies of distinguisher effectiveness to inform the design of CRSes.

Nearly all CRSes assume that some CRS components are located outside the censor's sphere of influence. However, recent disclosures by Edward Snowden call such design choices into question [93]. The alarming 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 basic CRS assumptions with respect to the censor's sphere of influence and visibility because existing CRSes are not designed to withstand global passive adversaries.

2.7.3 Security Gaps

Our analyses in Sections 2.4.5 and 2.5.3 reveal certain gaps in the security properties provided by existing CRS schemes. First, there are no effective countermeasures against poisoning of CRS information (*corrupt routing information* in Section 2.1.4). However, there is an implicit level of defense if the CRS uses public information; poisoning such information could cause collateral damage that may be unacceptable to the censor.

Second, most CRSes are susceptible to DoS attacks on key CRS components and resources [94]. CRSes implicitly depend on the capabilities of CRS participants or the provider that hosts the dissemination and CRS servers to prevent such attacks. Denial of service is a broader security issue: such attacks do not yet have a robust solution beyond over-provisioning of bandwidth and IP addresses.

Third, CRSes do not adequately defend against content corruption and malicious CRS participants; this is an area of active research [31] [66]. At present, this problem is primarily being addressed in the context of Tor ecosystem. Tor employs digital signature schemes to protect against poisoning of information about public relays, but authentication and confidentiality of bridge addresses are outstanding problems. There have been many DoS 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 a range of network-level tests, but this is done in an ad-hoc fashion.

2.7.4 Considerations for Participation

We note that most schemes in Communication Establishment (Table 2.1) and some schemes in Conversation (Table 2.2) recruit individual entities that volunteer to help based on *qualitative incentivization* (e.g. goodwill). Most CRS schemes treat participation as a binary decision; as a result, participants lack the ability to control the degree of cooperation (e.g. bandwidth, number of connections, or users). In most schemes it is not even clear what is the cost of participation. These issues may act as a deterrent to wide-scale CRS adoption among potential participants. A possible mitigation is to enforce a lower threshold on participation, giving participants the flexibility to switch between higher values and the minimum threshold. For example, Tangler [65]) demands a minimum amount of bandwidth or storage from participants.

Our study highlights an increasing trend among CRSes to leverage popular commercial services as participants (e.g. CloudTransport [95] hides user traffic by tunnelling it through Amazon's cloud storage [96]); this is particularly true of mimicry, tunnelling, and covert channel schemes which altogether represent 32 of the total 73 CRSes surveyed (Table 2.2). These implementations meet CRS design goals through happy and possibly temporary coincidence. First, such CRSes are contingent on participation of external parties that may not be invested in goals of the CRS. Second, the properties of the commercial party that are leveraged by the CRS do not exist by design. The third party may transition to unexpected states that break CRS functionality, or the assumed properties may simply disappear with an update or for commercial reasons. For example, Skype used to have supernodes (stable high-bandwidth nodes that relayed traffic between Skype users); supernodes were phased out in 2013 for scalability reasons. This affected mimicry-based CRSes (e.g.SkypeMorph [59]) that relied on Skype supernodes for privacy.

CRSes tend to entwine their operations with popular commercial parties assuming that the censor, being unable to accurately extract CRS activity, will refrain from wholesale blocking of the commercial party because of the collateral damage. Effectively, the CRS uses the commercial party as a concentration point to maximize the collateral damage incurred by the censor. Counterintuitively, the censor may actually benefit from this concentration; the attack surfaces and CRS security failures are also concentrated and well defined. For example, CloudTransport [95] uses traffic to only Amazon's cloud storage which makes it potentially easier to contain. A related issue is that the CRS operations may be disrupted if the censor develops local alternatives for the commercial parties employed by the CRS, forcing users in its sphere of influence to use local services. For example, during 2002–2005 access to `google.com` was slow and unreliable in China—requests for Google were blocked or redirected to local search engines with strict censorship policies. Prompted by these difficulties and losing business to its major Chinese competitor, Baidu [97], in 2006 Google opened its office in China and started `google.cn` that complied with China's censorship policy [98].

Finally, commercial participants are often operated by single corporations; the censor

can strike back at the CRS by attacking the platform or the entity that supports it. Such a strike can be in the form of actual network-level attacks [99] that cause the operator to reconsider or decry [100] its role as a host to CRS activity [101]. For example, four years after starting `google.cn` in China, in 2010 Google announced that it had faced sophisticated cyber-attacks from China causing loss of intellectual property for Google and unauthorized access to the email of dozens of human rights activists connected with China [102]. Consequently, Google started redirecting users visiting `google.cn` to its servers in Hong Kong and stopped censoring its search results. There is uncertainty around exactly how the third-party may respond under such circumstances: in the worst case it may even act as an informant to the censor and monitor CRS activity.

2.8 Summary

This chapter presented a comprehensive censor's attack model and established an evaluation framework for measuring security, privacy, performance, and deployability of censorship resistance systems (CRSes). We provided a comparative evaluation of existing censorship resistance approaches to emphasize their strengths and limitations. In a broader context, this study has highlighted a number of open areas and challenges: *(i)* combining CRSes may enhance security and privacy, but doing so in a meaningful way without breaking other operational assumptions is challenging, *(ii)* common CRS assumptions about the censor's sphere of influence and the cost of blocking incurred by the censor should be reevaluated in the light of evolving sociopolitical dynamics, *(iii)* there are outstanding security issues that CRSes cannot yet effectively mitigate such as denial of service attacks, corruption of information that supports CRSes (e.g. DNS poisoning), and malicious CRS participants, *(iv)* recruitment of volunteer participants should be at least partially regularized to ensure system stability and scalability, and *(v)* the growing CRS trend to rely on powerful commercial participants to increase the collateral damage caused by blocking can potentially act as a single point of failure from a security and privacy standpoint.

Chapter 3

The Consequences of Internet Censorship

In this chapter we discuss the consequences of user-side censorship, where some network device between users and publishers (e.g. state-level censor) blocks users' communications. Internet censorship artificially changes the dynamics of resource production and consumption, affecting a range of stakeholders. We provide quantified insights into the impact of censorship on users, content providers, and Internet Service Providers (ISPs) in the context of two large-scale censorship events in Pakistan: blocking of pornographic content in 2011 and of YouTube in 2012 (Sections 3.5, 3.6, and 3.7, respectively). Section 3.1 provides background on Internet infrastructure and censorship in Pakistan and also shows how this research relates to previous work. The datasets we draw upon for this study comprise six traffic traces collected before and after the porn and YouTube censorship events, including one dataset collected on the day when Pakistan blocked YouTube. We analyze this data to: *(i)* investigate changes in user behaviour (e.g. with respect to circumvention) after censorship, *(ii)* quantify the demand for blocked content and assess benefits extracted by competing content providers of blocked content, and *(iii)* illuminate challenges encountered by ISPs in implementing the censorship policies. This perspective is supplemented with a survey of about 700 Internet users in Pakistan (conducted one year after our last trace was captured). Section 3.2 describes our vantage point, data sources, and the data analysis process. To analyze the network traffic traces, we develop methodologies to establish censorship ground truth (i.e. what was censored and how it was censored) in Section 3.3.

3.1 Background and Related Work

This section provides context on the censorship events that we investigate and the relationship between our work and prior research.

Internet Infrastructure and Censorship in Pakistan. Our data spans network traces collected at an ISP in Pakistan between 2011 and 2013—a timeline during which

the country's censorship policy evolved. There are approximately 50 local and regional ISPs in Pakistan [103]. Only two of these, Pakistan Telecommunication Company Limited (PTCL) and Transworld Associates (TWA), have direct international connectivity which they sell to the rest of the providers as well as directly to consumers. Note that most Content Delivery Network (CDN) servers are located outside Pakistan. Internet censorship in Pakistan has mostly targeted content hosted outside the country, which Pakistani users access through PTCL or TWA.

The directives to block a particular website originate from the government or the judiciary. The ISPs are directed by the regulator, Pakistan Telecom Authority (PTA), to implement a content-blocking policy. While Pakistan has been intermittently blocking content since 2006 [104], a more persistent blocking policy was implemented in 2011 with the censorship of porn content [105], and then in 2012 with the blocking of YouTube [106]. The porn block in Pakistan was instituted in response to a media report that highlighted Pakistan as the top country in terms of searches for porn content [107]. The YouTube ban was triggered by a blasphemous video hosted on the website. When this study was conducted, the country continued to block access to YouTube as well as to websites deemed pornographic, anti-religious, or a general threat to national values and security [108]. Additionally, content related to human rights, independent media, proxy and circumvention tools, and BitTorrent file-sharing sites might also have been blocked [109].

Censorship Implementation. A large body of prior research infers censorship technologies by requesting potentially censored content to trigger censorship. These responses are then compared with baseline responses collected in uncensored regions. The goals of such studies are to detect the manipulation of traffic by intermediate devices [110] [111] [112] [113], to illuminate the nature of censored content [114] [115], and to uncover the mechanisms of censorship [116] [117] [118] [119]. Some recent studies analyze censorship in Pakistan [104] [109], finding ISP-level DNS redirection, HTTP-redirection, and fake-response injection at the national backbone as the mechanisms of censorship. We use previous work to validate our findings, but do not directly map it to our three-year dataset because censorship mechanisms can vary over time and across different vantage points. We perform passive analysis of individual data traces to identify the censorship mechanisms in effect at a given time. We are not aware of any prior work that reconstructs censorship mechanisms by passive analysis of network traces other than within the broader context of detecting forged TCP RST packets [120].

Consequences of Internet Censorship. Previous work examines how anti-piracy laws can affect the behaviour of users [121] and content providers [122], and investigates the (sometimes unintended) impact of Internet censorship on global Internet services. The injection of forged DNS responses by the Great Firewall of China (GFW) has been reported to cause a high level of collateral damage because it ends up also blocking outside

traffic that traverses Chinese links [46]. Upstream filtering can block traffic from outside a censored region because of ISP routing arrangements; for example, users of an ISP in Oman were unable to access certain content due to filtering regulations in India in 2012 [123]. Chaabane *et al.* analyze logs from Syrian censorship proxies to understand censorship methodology and circumvention trends [124]. Labovitz investigates how the takedown of MegaUpload servers in North America in 2012 affected file-sharing traffic [125]. The incident caused a small decrease in MegaUpload's previous traffic share, but made content delivery inefficient; files were fetched from European servers over more expensive transatlantic links.

For our purposes, a limitation of previous studies is that these employ vantage points that do not provide visibility into the full exchange of traffic between users and providers. Our study leverages an ISP viewpoint to investigate the consequences of Internet censorship on users, content providers, and ISPs. To the best of our knowledge, the last perspective has not been previously studied.

3.2 Data Sources for the Study

Our primary data consists of six network traces captured at a Pakistani ISP[1] between 2011 and 2013. As discussed in Section 3.1, the government of Pakistan implemented two significant and persistent censorship policies during this period. Figure 3.1 illustrates the temporal relationship of data capture dates to censorship events. The traces provide both pre-censorship and post-censorship snapshots of activity seen at an ISP corresponding to two major censorship events. We note that our data is not necessarily representative because it corresponds to just one ISP. Furthermore, it is difficult to estimate the actual user population of the ISP because of wide usage of network address translation (NAT) devices.

This data is supplemented with a user survey that we conducted in the region to explore user behaviour after YouTube was blocked. The survey results help to shape the scope of our analysis of YouTube censorship and provide an additional perspective for our findings from the primary data.

3.2.1 Capture Location and ISP Overview

We collected data at a tier-2 ISP in Pakistan that peers with a tier-1 provider through the Transworld Associates TWA-1 submarine telecommunications cable in Karachi. The ISP caters to both residential and Small Office/Home Office (SOHO) customers. Due to our confidentiality agreement, we cannot provide details about the scale at which the ISP operates, the magnitude of its customer base, or its address space.

[1] The ISP requested to treat its name, location, and other identifying information as confidential. The ISP acquired the data for unspecified purposes and provided a degree of access out of good will.

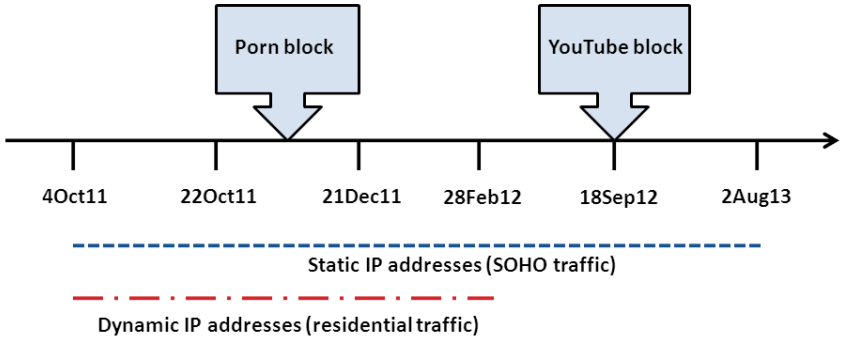


Figure 3.1: Temporal relationship of data to censorship events.

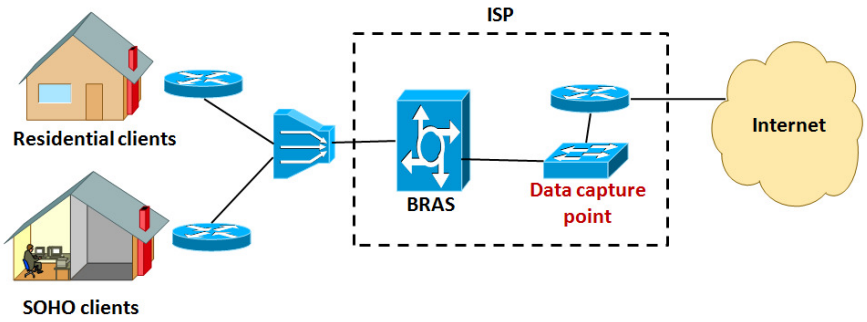


Figure 3.2: Data capture location.

Figure 3.2 shows the data capture location within the ISP premises. All customer lines terminate at one of several Broadband Remote Access Servers (BRASes) in the ISP's network. Each BRAS connects to the ISP's core Internet-facing router through a switch. The ISP captured data at the BRAS-facing side of this switch. This vantage point captures all the local ISP-generated traffic (e.g. redirected DNS traffic) as well as bi-directional traffic going in or out of the ISP's premises. The ISP assigns a set of addresses to each BRAS. This allocation remains unchanged for the duration of individual data traces; but across traces, potentially different subsets of the ISP address space might be in use.

Address Pools. The ISP splits its address space into dynamic DHCP and static pools. Residential customers are primarily assigned dynamic IPs. The ISP reserves some static IP addresses for hosting its services such as DNS resolvers, mail and authentication servers, and other Web resources. The bulk of the remainder is allocated to SOHO customers. We do not know details about the ISP's censorship apparatus, censorship blacklists, and IP addresses of the services hosted by the ISP. Note that the ISP does not allocate IPv6 addresses to its customers. Although our analysis revealed some IPv6 communication taking place over tunnels, its overall volume is negligible.

Block Key	Trace	Day	Capture Hour (PKT) + Duration	Size (GB)	Active Local IPs
–	03Oct11	Tue	17:48 + 15h14m	222	1,075
–	22Oct11	Sat	18:49 + 20h42m	460	1,046
P	21Dec11	Wed	22:17 + 16h54m	286	868
P	28Feb12	Tue	18:48 + 11h08m	200	974
PY	18Sep12	Tue	08:54 + 07h19m	500	310
PY	02Aug13	Fri	09:40 + 06h00m	207	136

Table 3.1: Summary of packet captures. P=Porn, Y=YouTube

3.2.2 Data Description

Table 3.1 shows a characterization of our data traces. Capture durations are 6 to 16 hours, and capture sizes are 200 to 500 GB comprising traffic from 100 to 1,000 IP addresses. Some of these IP addresses correspond to NAT devices; consequently, the effective user population could be larger than reported. A limitation of our work is that we cannot exclusively attribute cross-trace trends to the consequences of censorship; these might instead arise because of factors introduced by disparate capture days or timings. However, some of our results are strong enough to imply that these are prompted by censorship. Note that the characteristics of individual traces might differ from each other because the allocation of IP address prefixes to BRASes does not necessarily remain consistent across traces.

Protocol Logs. Our analysis is based on protocol logs generated from network traces using `Bro` [13]. Specifically, we process connection, HTTP, and DNS logs. The connection log contains one entry per flow, while the protocol logs contain entries for each request-response pair. We use *number of connections* to refer to distinct transport layer flows and *number of requests* for individual request-response pairs as observed in the protocol logs.

3.2.3 Data Sanitization and Characterization

We first identify measurement ambiguities and inaccuracies (*sanitization*) and then label data (*characterization*) to enable extraction of data subsets that are relevant to different analyses (Table 3.2). Various factors arising from limitations of the capturing device or analysis tool can introduce inaccuracies into data. A large portion of our data reflects connections that did not fully establish, possibly representing scanning activity. We remove such connections from the bulk of our analysis (we include unestablished connections in our analysis of user attempts to access blocked content). We only include connections for which a three-way TCP handshake was completed; this step reduces the number of connections seen across our datasets by half.

Block	Trace	Total	After sanitization	Transit	Local	Static IPs		Dynamic IPs	
Key		Conns.	(% retained)			Inbound	Outbound	Inbound	Outbound
–	03Oct11	11.53M	5.39M (46.7%)	0.03M	0.68M	1.05M	1.62M	0.54M	1.48M
–	22Oct11	29.19M	12.68M (43.4%)	0.03M	1.24M	3.58M	4.13M	1.25M	2.44M
P	21Dec11	16.06M	8.09M (50.4%)	0.02M	1.21M	1.37M	2.57M	0.50M	2.42M
P	28Feb12	12.12M	5.84M (48.2%)	0.04M	0.59M	0.86M	0.98M	0.99M	2.39M
PY	18Sep12	24.01M	14.93M (62.2%)	0.02M	2.19M	1.13M	11.59M	-	-
PY	02Aug13	8.79M	3.77M (42.9%)	0.01M	0.53M	0.19M	3.03M	-	-

Table 3.2: The breakdown of data sanitization and characterization. P=Porn, Y=YouTube

We label flows based on the connection direction and the type of local addresses to facilitate extraction of data subsets suitable for different analyses (e.g. to assess the interest of local users in Web content, we consider outbound connections). We label a connection as *local* if it has both source and destination IP addresses in the ISP's network block or *transit* if neither source nor destination IP addresses belongs to the local network. A connection is labelled *inbound* if its originator resides outside the ISP's network and *outbound* in the reverse case. Using the information (provided by the ISP) that nearly all static IP addresses correspond to SOHO users, we label traffic as SOHO and residential.

Table 3.2 summarizes these characterizations. We find that a large number of connections are outbound, followed by inbound, local, and transit connections, respectively. The small portion of transit traffic agrees with a communication from the ISP that we should expect a small amount of traffic from a sister ISP and some IPv6 test traffic.

3.2.4 Final Datasets

Table 3.3 summarizes the filtered dataset. We divide the six traces into two datasets corresponding to residential and SOHO users. We use both datasets to study the consequences of porn censorship, and only the SOHO dataset to analyze YouTube censorship (because the fraction of residential traffic in traces collected after YouTube block is negligible). We characterize outbound and local connections in HTTP and DNS logs. Note that our analysis includes local traffic because we expect that a portion of user traffic will be redirected to the ISP's censorship machinery.

3.2.5 User Survey

We carried out an online user survey targeting users in Pakistan to understand their perceptions about the YouTube block. We avoided asking questions about the porn block as it is a culturally sensitive topic in the region. The survey included questions about *(i)* the popularity of blocked content and new players that emerged post-censorship, *(ii)* user inclinations to circumvent and the corresponding mechanisms, *(iii)* the collateral damage experienced because of the block, and *(iv)* opinions about Internet censorship in general. The survey is included in Appendix B. We disseminated information about the opportunity to take the survey through mailing lists and classroom discussions in Computer Science and Engineering departments in two universities in Islamabad and Lahore. We did not expect many responses because it is hard to get users to respond to surveys without any incentive, especially when the topic is a sensitive one such as Internet censorship. Surprisingly, we received 770 responses, reflecting a widespread eagerness to comment on the subject.

Block Key	Trace	Active IPs	Conns.	TCP Conns.	UDP Conns.	HTTP Transactions	SSL Conns.	DNS Conns.	Bytes (GB)	Packets
SOHO Traffic (Static IPs)										
–	03Oct11	585	2.02M	1.00M	1.02M	1.44M	0.05M	1.29M	79	119M
–	22Oct11	554	4.84M	1.91M	2.93M	2.18M	0.09M	1.90M	180	276M
P	21Dec11	570	3.24M	1.70M	1.55M	2.52M	0.14M	2.63M	121	182M
P	28Feb12	298	1.16M	0.51M	0.65M	0.62M	0.08M	0.33M	39	61M
PY	18Sep12	298	13.78M	7.53M	6.25M	7.16M	1.05M	4.26M	271	546M
PY	02Aug13	133	3.56M	1.85M	1.71M	1.78M	0.32M	1.57M	143	246M
Residential Traffic (Dynamic IPs)										
–	03Oct11	490	1.76M	0.85M	0.9M	1.14M	0.05M	1.86M	85	149M
–	22Oct11	492	2.97M	1.40M	1.57M	1.84M	0.08M	1.08M	163	237M
P	21Dec11	451	2.96M	1.50M	1.45M	2.11M	0.13M	1.09M	103	176M
P	28Feb12	676	2.80M	1.26M	1.55M	1.46M	0.11M	0.80M	112	176M
PY	18Sep12	-	-	-	-	-	-	-	-	-
PY	02Aug13	-	-	-	-	-	-	-	-	-

Table 3.3: The final data after preprocessing. P=Porn, Y=YouTube

We note that the demographic of survey participants does not reflect Pakistan's makeup as a whole, and is likely skewed towards particularly informed and active users because of the methodology of survey dissemination. Participants were predominantly male (75%), possibly reflecting female underrepresentation in Computer Science and Engineering fields in general. Majority of the participants resided in urban areas, mainly living in Islamabad (33.6%) and Lahore (46.3%). Our form included questions about participant age and occupation as optional questions. Only 47 participants provided their age (of which 28% were in the range 20–29 years, 23% in 30–39, 15% in 15–19, 13% in 40–49, 13% in 50–59, and 8% in 60–65), and only 150 participants shared their occupation (41% of which were University Students, 37% Computer Scientists/Engineers, 13% University Faculty, 3% Medical Doctors, 2% University Admin, 2% business and sales, 1% homemaker, and 1% indicated 'Other' profession). Clearly, the survey has response bias because of the dissemination methodology. Additionally, respondent age and occupation are inconclusive because of a flaw in survey design that led respondents to treat these questions as optional. Therefore, we do not frame the results of this survey as representative, but rather as illuminating some facets of how censorship affects Pakistani users—with due consideration for all the survey limitations previously noted.

3.2.6 Ethical Standards

This work involves analysis of data directly obtained from human subjects via user survey, and indirectly by analyzing user-generated traffic captured by an ISP. We discuss ethical considerations that were employed to limit potential harm in the handling of data, and to balance probable harm and societal benefit resulting from the research [126] [127].

The ISP gave the author of this dissertation direct access to the data via SSH over VPN. She acquired permission from the Ethics Committee of the University of Cambridge, providing a description of the nature of the protections used to limit potential harm in the gathering, storage, analysis, and presentation of data collected at the ISP as well as the user survey. She signed a contract with the ISP that obligated her to: *(i)* respect user privacy, *(ii)* not share data with third parties (including co-authors), *(iii)* not move data outside Pakistan, *(iv)* not move data within Pakistan without prior consent, and *(v)* undertake an objective study and refrain from maligning any party (users, ISPs, or the government). These restrictions did not affect the accuracy of our results.

The network data used in this study cannot be used to identify and potentially harm individuals. The ISP provided anonymized network logs where IP addresses had been hashed to protect privacy of its users. Furthermore, the results of this study are based on aggregate analysis of traffic trends pre- and post-censorship, which cannot be mapped to individuals. It is not known if the ISP acquired consent of its users. However, we believe that the insights regarding the nature and effects of censorship provided by aggregate analysis of data outweighs the potential absence of user consent, which is infeasible to acquire post hoc as is often the case with leaked historic data.

We sought and were granted approval by the two universities in Pakistan through which the survey was disseminated in Pakistan. Contact details of our references in Pakistan from whom permission was obtained to disseminate the user survey was shared with the Ethics Committee of the University of Cambridge. The survey can potentially compromise participant safety through identification and subsequent prosecution by the censor. The survey has been designed to not elicit any identifying information from participants. Moreover, the survey was created using Google Forms which uses encrypted connection, ensuring confidentiality of responses. It is possible for Google to expose survey participants voluntarily or through coercion by sharing information with the government. However, we believe that the risk is low as a growing number of companies are developing strategies to resist pressure from governments to disclose private user information [8].

As noted by Wright *et al.*, many nations have loosely-defined computer crime laws: the legal status of attempting to access blocked content or the use of circumvention tools is not entirely clear [128]. Conversely, user practices regarding censored content that technically violate the law might not jeopardise their safety in practice. Pakistan is relatively tolerant of citizen views about censorship. Social welfare organisations in Pakistan such as Bolo Bhi openly campaign against Internet censorship in the country [129].

While establishing the censor's blocking mechanisms, this study highlights limitations of these mechanisms and incomplete coverage of the porn blacklist. The censor can potentially use results from this study to improve its censorship strategy. However, mechanisms of censorship and their limitations are well-known (Section 2.1.4). The censor can potentially acquire relevant expertise from other countries—but as we discuss in Section 2.1.2, such decisions are subject to physical, economical, technological, and political constraints of the censor. We believe that this study deepens our understanding of censorship by showing its evolution and effects in the context of Pakistan, which outweighs the potential risk of helping the censor to enhance its censorship strategy.

3.3 Establishing Ground Truth

A significant challenge for our study is that we use historical data for which contextual information—*what* was censored (the blacklist for the porn block) and *how* it was censored (the mechanism of censorship)—is not available. In this section we discuss the methodology that we developed to answer these questions based solely on the information present in our data traces. We analyze server responses to user requests, basing our deductions on the observation that for enforcing censorship, a censor either silently drops requests or sends back false response packets.

3.3.1 Censorship Indicators

A censor can block HTTP content at any of the layers involved in an HTTP transaction: DNS, TCP/IP, and HTTP. Across these layers, the censor has an array of choices for how to block, each leaving a trail in the network traces. The presence of such a trail—a sequence of packets (not necessarily contiguous) or the absence of expected packets—provides an indicator of censorship. However, some of these indicators can show up in an uncensored environment for legitimate reasons such as measurement loss or excessive server load. We deem such censorship indicators as *ambiguous* and handle these as follows: *(i)* if the censored content is *known*, we attribute a high frequency of an ambiguous indicator to censorship (and leverage this information to establish the mechanism of censorship), and *(ii)* if the censored content is *unknown* (i.e. we cannot associate flows with attempts to access blocked content), we cannot attribute the occurrence of such an indicator over a short observation window (less than one day for all our traces) exclusively to censorship. We rely on only unambiguous indicators to establish (partial) ground truth. We now discuss assessing censorship indicators at each layer.

DNS-Based Censorship. A DNS resolver controlled by the censor (such as the one maintained by the ISP) can effect blocking by sending responses that can be classified as: *(i)* No Response, *(ii)* False Error (e.g. NXDOMAIN), or *(iii)* False Response (the code for such responses is NO ERROR). (Clearly, users can bypass DNS-based censorship by using an independent DNS resolver.)

No Response. This is an ambiguous indicator because it can occur because of network problems or excessive load on the resolver. We do not attribute this scenario to censorship when the censored content is unknown. However, for known censored content, observing a consistent No Response behaviour is a strong indicator of censorship.

For False Error and False Response, we can leverage two public databases to establish the ground truth: *(i)* `dnsdb` maintains historical data containing name to IP address mappings [130], and *(ii)* Team Cymru's database of IP to ASN mappings [131]. We now describe our methodology to detect false responses.

False Error. We flag domains in queries that consistently receive an error response code from a DNS resolver for a subsequent `dnsdb` lookup. If there exists a name-to-IP mapping in the database for the flagged domains, we infer that the censor employed False Error as the mechanism of censorship.

False Response. We detect False Response by analyzing if a DNS resolver consistently provides 'similar' IP addresses in its DNS responses. We flag the cases in which a DNS resolver's answer contains an IP address that belongs to an ISP within the country (i.e. the local ISP or an upstream transit provider within Pakistan) but the domain is actually hosted outside Pakistan.

Let $\text{ASN}_{\text{trace}}$ be the ASN of an IP address returned in a DNS reply recorded in the trace, and ASN_{real} be the ASN of the IP address received in a DNS reply obtained by

actively resolving the same query through an uncensored DNS resolver. If $\text{ASN}_{\text{trace}}$ belongs to an ISP within Pakistan, but ASN_{real} does not, we infer that the query received a false response. A limitation of this technique is that we cannot detect cases where the censor redirects DNS queries to an IP address that belongs to an AS outside the country. Furthermore, this technique flags caching servers employed within the country.[2]

TCP/IP blocking. IP blocking is an ambiguous indicator of censorship because it is hard to distinguish from legitimate causes of inaccessibility. However, we can weed out a fraction of the non-censorship cases based on the assumption that in the case of censorship *all* attempts to establish a connection with a blocked address will fail. First, we collect IP addresses from `A` records for all the correctly resolved queries. Next, we label connections matching these IP addresses as belonging to one of the following three connection states.

- **PARTIAL.** We did not observe SYN from the connection originator, but packets from the responder were seen.
- **EST.** A full TCP establishment handshake was observed.
- **BLOCKED.** The originator sent a SYN, but either it receives no response or receives a RST (potentially injected by the censor).

We flag IP addresses for which we never observe EST and for which we observe BLOCKED at least once. We map these IP addresses back to the corresponding domain names in the DNS logs and consider these domains as potentially censored.

HTTP-Based Blocking. A censor can effect HTTP blocking by: *(i)* sending no HTTP response (e.g. by injecting a RST after connection establishment), *(ii)* returning an HTTP error response code, or *(iii)* returning a false response such as a block page (either directly or through HTTP redirection). We examine these cases as follows.

- **No Response.** An HTTP request can receive no response because of legitimate reasons. We do not attribute this case to censorship if the censored content is unknown. For known censored content, consistently observing TCP blocking indicates censorship (e.g. if the responder always sends RST in response to an HTTP request).
- **Error Response Codes.** This is an ambiguous error because a client can legitimately receive such responses if the requested resource is forbidden or not found. However, for known censored content, this provides a strong indicator of censorship if HTTP errors are the dominant behaviour.

[2] We identify the ISP's caching machines using this methodology.

	Trace	DNS	IP	HTTP
YouTube	18Sep12	DNS_REDIR	—	HTTP_REDIR
	02Aug13	DNS_REDIR	—	HTTP_NORESP
Porn	21Dec11	DNS_REDIR	—	—
	28Feb12	DNS_REDIR	—	—
	18Sep12	DNS_REDIR	IP_BLOCK	—
	02Aug13	DNS_REDIR	—	HTTP_NORESP

Table 3.4: Censorship mechanisms for YouTube and porn blocking as observed in our post-censorship traces. "—" indicates that we do not find any conclusive evidence of the given mechanism.

- **Block Page via 3XX redirection.** The censor may redirect blacklisted domains and sub-domains to a *Location* controlled by the censor. We can detect this mechanism by analyzing histograms of the value of `Location` header in HTTP responses assuming that the censor redirects to a small set of locations. However, if the censor redirects HTTP requests for censored content to distinct locations (e.g. by incorporating URI in the HTTP request into the redirection location), the histogram will not reveal common redirection targets.

- **Block Page via 2XX response.** It is common for censors to display the same block page for all blacklisted URLs. This behaviour can be detected by fingerprinting block pages associated with the censor. For example, block pages can be identified by examining spikes or modes in a histogram of the number of bytes sent in HTTP responses.

3.3.2 Mechanism of YouTube Censorship

Table 3.4 shows the mechanisms of YouTube censorship identified by our study in the two traces corresponding to the YouTube block. We find that the ISP's DNS resolvers redirect queries for YouTube to a single IP address owned by the ISP, but non-ISP resolvers give correct answers. We do not find any instances of IP blocking (there is only one potentially blocked address in 18Sep12, which reverse-maps to a YouTube content server).

We also observe HTTP blocking in both traces. In 18Sep12, we find blocking of YouTube via 3XX redirection to an IP address owned by a large local provider—one of the two providers with direct international connectivity. In 02Aug13, the blocking mechanism shifted from HTTP redirection to No Response. In traces before 02Aug13 (including traces captured before the YouTube block), approximately 2% of HTTP requests receive no response; whereas in 02Aug13 this fraction jumps to approximately 95%, with nearly all such connections receiving RST from the responder.

Trace	DNS	IP blocking	HTTP blocking
21Dec11	226	3 / 0%	2 / 0%
28Feb12	145	7 / 0%	1 / 0%
18Sep12	105	56 / 41%	6 / 0%
02Aug13	100	0 / 0%	8 / 62%

Table 3.5: The number of porn domains potentially blocked at each layer. For IP and HTTP blocking, we also show the percentage overlap with DNS blocking. HTTP blocking when present takes the form of consistent No Response conditions.

Porn domains	Oct11	Dec11	Feb12	Sep12	Aug13
Unblocked	1,313	1,181	1,609	2,210	2,352
Blocked	0	226	145	161	105
New entries	—	0	37	36	0
% overlap	—	8.2%	0.2%	0.5%	0%

Table 3.6: The evolution of porn blacklist as seen in our traces. % overlap corresponds to the proportion of new entries present (and unblocked) in all previous traces. Oct11 reflects the two traces captured in October 2011.

These observations confirm the two-layered censorship mechanism described by a prior study [104]: ISPs locally enforce blocking via DNS redirection, and the two large providers in Pakistan with direct international connectivity (PTCL and TWA) employ HTTP blocking.

3.3.3 Mechanism of Porn Censorship

We characterize all websites recorded in our traces using McAfee's URL categorization service [132], extracting the ones that are classified as *Pornography*. We spot-checked a random sample of the service's decisions (both positive and negative) to confirm its apparent accuracy and lack of any regional variations: we did not find any errors.

We recover the censor's *porn blacklist* for each trace individually. Note that the blacklists represent only a fraction of the censor's true blacklist as our data-driven approach can only identify the censored content present in our traces.

We aim to recover blacklists at the granularity of *registered* domains as per the master list kept by Mozilla [133]; later in our analysis, we use this granularity to characterize traffic to blacklisted domains. Where applicable, we mark domains in our blacklist as partially blocked: we consider the possibility of partial blocking of a domain only at the IP layer (due to incomplete IP address coverage) and at the HTTP layer (due to incomplete

regex coverage). For DNS blocking, we assume that the true blacklist contains domains at the granularity of *registered* domains. For this reason, we only add a domain to the blacklist if we observe consistent blocking behaviour for all of its subdomains that appear in a given trace.

Table 3.4 summarizes the mechanisms of porn censorship in the four traces relevant to porn censorship and Table 3.5 shows the corresponding development of the porn blacklist. We make the following observations.

- We find evidence of DNS redirection in all the four traces. The ISP's DNS resolvers consistently redirect queries for blocked domains to an IP address owned by the ISP (queries for YouTube are also redirected to this IP address). However, non-ISP resolvers correctly resolve queries for blocked content, indicating that the censor does not employ DNS injection such as discussed by Duan *et al.* [134]. Table 3.5 lists the number of blacklisted domains per trace that we recover using this indicator.

- We observe IP blocking for some porn domains. Because this is an ambiguous indicator, we examine the overlap of censored domains found via IP blocking with those found via DNS blocking. We find significant evidence of IP blocking in 18Sep12, which has 41% overlap with the DNS blacklist. The TCP states of these connections indicate that the originator did not receive any response packet from the responder, consistent with blackholing.

- We do not find any instances of users receiving an HTTP block page, whether through injection or redirection. Some domains consistently receive no HTTP response, but with negligible overlap with our DNS blacklist (except for the last trace) as shown in Table 3.5. The TCP states of these connections reveal that in most cases the responder terminated the connection by sending a RST.

We do not find concrete evidence of extensive IP- or HTTP-level blocking of porn, except for the cases where we observe a high overlap with the DNS blacklist. Consequently, we do not include these ambiguous domains in blacklist reconstruction, resulting in the omission of only a handful of potentially blocked domains.

Table 3.6 illustrates how the porn blacklist evolved over time. We consider the question whether the censor acts in a reactive fashion; that is, does the censor block porn domains that begin to gain popularity with users? In pre-block traces (03Oct11 and 22Oct11), we see 1,313 unique porn domains. We find that 8.2% of these domains were blocked in 21Dec11. After the initial dissemination of the blacklist in 21Dec11, we see a lull in its update behaviour: we observe only approximately 35 new domains added in each of 28Feb12 and 18Sep12, and no new domains in the last trace. Moreover, the 'new' blocked domains have only a small overlap with the unblocked porn domains observed in prior traces—reinforcing the information unofficially shared with us by local operators that blacklists are disseminated to ISPs by the central censorship regulator, and that

the development of these blacklists is independent of the porn browsing trends of users (Section 3.1).

Summary. Table 3.4 provides a summary of our findings on the mechanisms of YouTube and porn censorship. We find DNS blocking of both YouTube and porn via redirection to an IP address controlled by the ISP. In 18Sep12, YouTube is also blocked using HTTP redirection, and porn using IP blocking. Both YouTube and porn are blocked via RST injection in 02Aug13.

3.4 Metrics Relevant to Content Providers

In this section we discuss two key aspects for our study: *(i)* what constitutes a "content provider" relative to each censorship event, and *(ii)* the metrics on which we base our assessment of changes resulting from censorship events (note that we can only apply these metrics to unencrypted traffic).

Censorship events affect both primary and alternate providers of the censored content. For the YouTube event, these relate to the general category of *Video Content Platforms*, for which we focus our analysis on four major players: YouTube, DailyMotion, Tune.pk and Vimeo. These constitute the primary video providers for Pakistan as based on their market share [135] and the results of our user survey in Section 3.2.5. For the porn censorship event, we consider all porn domains seen in our traces as identified by McAfee's URL categorization service in April 2014. Given that our most recent trace was captured in August 2013, some domains might have been inaccurately classified.

The primary metric that we employ is *downstream* traffic (server response bytes) served by blocked and alternate content providers, which we will often abbreviate as "bandwidth" for shorthand. We base this choice on the observation that both the censored categories, video and porn, make heavy use of network downloads—what is censored primarily constitutes images and videos. For these categories, downstream bandwidth reasonably captures the degree of user interest in a content provider. This metric also allows us to readily study shifts in traffic trends in the presence of encryption technologies—a potential response to broad category-based censorship.

In addition, for the video category, we assess changes in content *embedded* in other sites in response to censorship.[3] This metric captures the broader ecosystem for users viewing videos sometimes in response to other websites that embed a content provider's videos. (Porn content, on the other hand, is presumably only embedded on other porn sites.) After censorship of a content provider, local websites lack an incentive to embed the provider's videos. We now discuss how to compute these two metrics.

[3] We treat links to a provider in search results as a form of direct access; we presume that often users navigate to such pages via search engines. For embedded content, we consider only those links where the content has been integrated into the Web page, for example Dramas Online is one such website that embeds videos of popular Pakistani drama serials from different TV channels [136].

Direct vs. Embedded Video Viewing Requests. To distinguish between these two types of requests, we need to develop *signatures* that classify a given URL as direct, embedded, or neither. One approach for developing signatures is to analyze traffic dumps collected by actively downloading video content [137]. However, given we collected our traces over a span of three years, we cannot employ an active approach like this, as signatures can change over the years. To develop signatures that can span our datasets, we use a data-driven methodology: for each video content platform, we examine a histogram of its URI root prefixes and associate them with distinct classes of Web content based on inspecting the corresponding content type observed in traffic captures, and in some cases entering the full URL into a browser to see if the video plays. This approach provides us with fingerprints for both *direct* and *embedded* viewing request URLs for each video content platform. We note that direct and embedded video watching requests have a consistent signature across traces, perhaps because these span the same domain.

Bandwidth per Content Provider. We could compute downstream bandwidth by accumulating server bytes for all HTTP requests where the content provider domain appears in the `Host` header. However, this approach risks missing traffic because: *(i)* content can be served by CDNs (often the case for videos and images), the domain name of which may have no relationship to the host corresponding to the original video/image request, and *(ii)* CDNs typically serve content on behalf of multiple domains, making it infeasible to exclusively associate a given CDN domain with a specific origin server. We might consider accounting for such traffic by accumulating all response bytes for requests where the content provider appears in the HTTP `Referer` header, but doing so will: *(i)* include bytes belonging to other websites or providers, since the `Referer` might instead reflect the user clicking on a link on the original content provider page that leads to a *different* content provider's page, and *(ii)* miss bytes belonging to the content provider in cases where an automatic chain of requests traverses multiple domains in order to ultimately reach the CDN.

Putting the above considerations together, we employ the following approaches for estimating traffic volume.

- **Video Content Platforms.** Because these analyses concern just a handful of content providers, for the video category it remains practical to develop URI signatures for each of the four major players. Note that these signatures are different from the *direct* and *embedded* watch signatures, because the video is in general fetched from a URL different than that of the watch page. Along with analysis of active fetches, we analyze all HTTP requests where either `Host` or `Referer` header contains the content provider's domain, and the `Content-Type` header in the response is either `application/octet-stream` or contains the keyword "`video`". It follows that we miss video downloads for content providers whose domain name appears in neither the `Host` nor the `Referer` part of an HTTP request. We find that YouTube

Key	Trace	HTTP GB	% Porn	% Video	HTTP:SSL
		SOHO Traffic			
–	03Oct11	58.15	11.5	45.5	40.72
–	22Oct11	105.79	11.6	53.6	38.19
P	21Dec11	90.05	3.7	50.2	23.72
P	28Feb12	23.37	2.0	54.3	17.77
PY	18Sep12	91.60	3.0	11.7	3.20
PY	02Aug13	49.66	3.8	5.5	3.25
		Residential Traffic			
–	03Oct11	52.10	9.4	—	20.05
–	22Oct11	100.04	7.4	—	50.30
P	21Dec11	66.70	4.0	—	18.22
P	28Feb12	66.23	3.5	—	14.33

Table 3.7: The ratio of porn and general video traffic to total HTTP byte volume. The last column shows the ratio of HTTP volume to TLS/SSL volume. "—" indicates a datapoint not considered in our study (we only use SOHO traffic for analyzing the YouTube block). P=Porn, Y=YouTube

transfers video using both `video` and `application/octet-stream`. The other three providers, however, only transfer video using a `video` content-type (and sometimes employ `application/octet-stream` for content such as CSS and fonts).

- **Porn Providers.** Accurately attributing porn bandwidth requires a more generic approach, since there are too many providers (approximately 3,800 seen in our traces) to allow us to craft individual signatures. Since a porn site can embed content from other porn sites, when we see a transfer for which both the `Host` domain and the `Referer` domain are labeled as porn in our dataset, we give priority to the former. Specifically, we use the rule: if `Host` has porn domain X, add the corresponding bytes to X; else if `Referer` exists and has a porn domain Y, add corresponding bytes to Y. Otherwise, do not attribute the transfer to any domain.

3.5 Changes in User Behaviour

In this section, we quantify user demand for blocked content before censorship, and their persistence and approaches in accessing blocked content after censorship comes into place. While we cannot rule out other factors leading to some of the changes we have observed,

Key	Trace	Total	YouTube (%)	Others (%)	Breakdown of Others		
					DailyMotion (%)	Tune.pk (%)	Vimeo (%)
Video Bandwidth (GB)							
–	03Oct11	26.5 GB	97.9	2.1	2.0	0.0	0.1
–	22Oct11	56.6 GB	97.6	2.4	2.4	0.0	≈ 0.0
P	21Dec11	45.2 GB	98.5	1.5	1.3	0.0	0.2
P	28Feb12	12.6 GB	96.9	3.0	3.0	0.0	≈ 0.0
PY	18Sep12	10.7 GB	15.8	84.2	82.0	0.0	2.2
PY	02Aug13	2.7 GB	0.0	100.0	40.9	57.6	1.5
Number of Direct Watch Requests							
–	03Oct11	2,199	99.5	0.5	0.2	0.0	0.2
–	22Oct11	4,550	99.0	1.0	0.9	0.0	0.0
P	21Dec11	3,254	99.3	0.7	0.6	0.0	0.1
P	28Feb12	878	95.7	4.3	4.3	0.0	0.0
PY	18Sep12	992	71.1	28.9	23.2	0.0	5.7
PY	02Aug13	169	46.1	53.8	37.3	14.2	2.4
Number of Embedded Watch Requests							
–	03Oct11	200	87.0	13.0	10.0	0.0	3.0
–	22Oct11	299	78.9	21.1	14.7	0.0	6.4
P	21Dec11	414	92.5	7.5	2.7	0.0	4.8
P	28Feb12	209	86.1	13.9	11.5	0.0	2.4
PY	18Sep12	2,037	73.0	27.0	19.5	0.0	7.5
PY	02Aug13	647	51.8	48.2	32.6	10.7	4.9

Table 3.8: The distribution of video bandwidth, and the number of direct and embedded watch requests across major video content providers over time.

the broad scope of the censorship events makes it quite likely that our observations indeed reflect responses to censorship.

3.5.1 Changes in Traffic

We observe in Table 3.7 that on average video traffic represents 50% of HTTP traffic before the YouTube block, consistent with global trends (a 2012 study found that 57% of user-generated traffic was video [138]). The overall (unencrypted) video consumption rate drastically declines after the YouTube block, subsequently constituting only 12% of total HTTP traffic in 18Sep12, and declining further to 5.5% in 02Aug13. The decline

in video traffic coincides with a decrease of nearly 90% in the HTTP to SSL[4] ratio in 18Sep12 (the day when YouTube was blocked). The ratio remains fairly consistent on this day as viewed hour-to-hour (on average about 3.25), indicating that SOHO users quickly switched to SSL-based circumvention technologies. The trace for this day does not reflect a clear learning phase, suggesting that this change had likely already occurred by the time the capture began. The overall trend for SSL traffic remained consistent 11 months later in 02Aug13. The steep increase in SSL traffic after the YouTube block suggests that most users switched to encrypted tunnels to watch video content. Our user study substantiates this observation: 57% of the survey participants state that they use SSL-based VPN software (i.e. UltraSurf, OpenVPN, and Hotspot Shield) to access YouTube.

Table 3.8 shows direct video requests for YouTube (via user navigation or mediated by clicking on search results). The majority of direct video requests prior to the block correspond to YouTube (average 98%). Immediately after the block (18Sep12), YouTube still receives the highest portion of direct requests (even though reduced by 27%). The proportion of YouTube traffic sharply drops 11 months later in 02Aug13 to 46%, as users begin to disperse their requests among alternate providers. The decrease in direct YouTube video requests matches our survey results: 40% of respondents do not click on YouTube links because of the block, 39% employ a circumvention mechanism to access the link, and 17% participants access the video via an alternate provider.

Table 3.7 shows that before the block, the average porn bandwidth is 8.4% and 11.5% of HTTP traffic to residential users and SOHO users, respectively. These numbers lie below global estimates that report porn to constitute 30% of Internet traffic [139]. SOHO users consume higher porn bandwidth than residential users, possibly because of the availability of higher bandwidth. After the block, the residential porn bandwidth falls by more than half (i.e. about 3.7% of HTTP traffic).

For SOHO traffic, the average porn bandwidth reduces by a factor of three. In contrast to video, we do not observe a significant increase in the HTTP-to-SSL traffic ratio in response to porn censorship, indicating that a subset of users either stopped watching porn or shifted to alternate porn providers.[5]

3.5.2 Effects on User Behaviour

Censorship can potentially modify user behaviour or result in new behaviour (e.g. attempting to bypass the block). In this section, we study temporal patterns in the use of DNS resolvers and Web proxies, and user response after encountering the block page.

[4] We cannot conclusively say if the SSL traffic corresponds to VPNs or HTTPS.

[5] In 18Sep12 and 02Aug13, the SSL ratio in SOHO traffic increases by several orders of magnitude. However, this is likely to be a consequence of YouTube censorship which spans this timeline.

Resolver ASN shorthand (% of DNS queries)					
–	–	P	P	PY	PY
03Oct11	22Oct11	21Dec11	28Feb12	18Sep12	02Aug13
SOHO Traffic					
39,248	64,269	43,655	10,062	13,025	5,036
Local-ISP (99.89)	Local-ISP (99.58)	Local-ISP (98.23)	Local-ISP (91.93)	Local-ISP (68.75)	Local-ISP (74.12)
Google (0.06)	Google (0.28)	Google (1.46)	Google (5.63)	Google (13.69)	Google (19.32)
LEVEL3 (0.04)	LEVEL3 (0.08)	LEVEL3 (0.12)	VPLSNET (1.37)	LEVEL3 (6.96)	LEVEL3 (2.88)
PKTELECOM-AS-PK (0.01)	IPC Computing (0.03)	VPLSNET (0.10)	HINET (0.96)	SPEEDCAST (5.51)	MULTINET (2.70)
VeriSign (0.01)	DIEGOGARCIA (0.02)	HINET (0.08)	OpenDNS (0.11)	OpenDNS (5.08)	Verizon (0.99)
Residential Traffic					
12,739	15,821	6,767	5,451	—	—
Local-ISP (95.50)	Local-ISP (93.89)	Local-ISP (93.08)	Local-ISP (92.20)	—	—
ASVPSHOSTING (3.73)	OpenDNS (5.49)	Google (6.18)	ASVPSHOSTING (4.59)	—	—
Google (0.69)	ASVPSHOSTING (0.49)	ASVPSHOSTING (0.62)	Google (3.05)	—	-
CELCOMNET (0.06)	LEVEL3 (0.11)	LEVEL3 (0.09)	LEVEL3 (0.13)	—	—
OpenDNS (0.02)	TIGGEE (0.01)	OpenDNS (0.03)	OpenDNS (0.04)	—	—

Table 3.9: The distribution of DNS A/AAAA queries for blocked categories across top 5 DNS resolvers. The top rows in Residential Traffic and SOHO Traffic sections represent the total number of DNS A/AAAA queries per trace for blocked categories.

DNS Resolvers. Sections 3.3.3 and 3.3.2 show that the ISP's DNS resolvers redirect requests for blocked domains to an IP address owned by the ISP. This block can be circumvented by using a non-ISP DNS resolver. Table 3.9 shows the degree to which users employed this circumvention technique by examining the top 5 DNS resolvers used to resolve DNS A or AAAA queries across the six traces.[6] We find that before censorship, the ISP resolves at least 90% of queries for both YouTube and porn. After porn censorship, we observe a small increase (about 5%) in the use of Google's public DNS resolvers accompanied by a decrease in the use of the ISP's resolvers. This number rises to about 13% after the YouTube block. At the same time, the queries resolved by the ISP's resolvers drop to 70%, and we see an increase in the queries resolved by OpenDNS and LEVEL-3. The use of alternate DNS resolvers to circumvent censorship has been noted in the context of other censorship incidents [140], and potentially increases user exposure to security risks [141].

Web Proxies. We extract domains from HTTP requests that are classified by McAfee's categorization service as *Anonymizers*. For SOHO traffic, we observe only 1 Web proxy prior to the YouTube block, which rises to an average of 41 proxies after the block, with a striking 114 proxies on the day of YouTube block. Residential traffic has the same distribution of Web proxies before the block as SOHO traffic, and a less dramatic increase of 11.5% on average in the use of Web proxies after the block. We also extract domains from SSL certificates, finding *no* proxy hosts before the YouTube block. After the block, we observe 15 and 8 proxy hosts in 18Sep12 and 02Aug13, respectively.

We manually inspect proxy hosts and find that these either offer encryption by default or provide easy options to enable encryption. For example, the top two (`youtubeproxy.org` and `12345proxy.net`) use HTTPS by default, and a popular host (`4everproxy.com`) prominently lists HTTPS-based proxies on its home page. In addition, the respondents to our survey indicated that SSL-based software such as OpenVPN and Hotspot Shield are among the most popular circumvention tools. Apparently, these tools became popular during the one year between our last trace and the survey; we did not find dominant trends for such tools in our data.

User Behaviour after Viewing Block Page. We develop insights into how users respond to censorship by analyzing their actions after encountering a block page; that is, do they attempt to access similar unblocked content, employ circumvention, or apparently give up by shifting to some other activity.

When a user encounters a block page, we analyze the user's HTTP transactions in the subsequent 5-minute window. To reduce ambiguities because of IP aliasing, we confine

[6] To mitigate bias caused by automated DNS queries that might use a diverse set of DNS resolvers, we limit our analysis to queries for the blocked categories because such queries are not likely to be generated by non-human actors.

Domain Shorthand (% of total porn bandwidth)					
03Oct11	22Oct11	21Dec11	28Feb12	18Sep12	02Aug13
Residential Traffic (GB)					
4.91	7.37	2.67	2.32	—	—
A / 42.3%	A / 26.4%	***I*** / 22.8%	***M*** / 23.2%	—	—
B / 12.1%	B / 15.4%	**J** / 16.8%	***R*** / 13.7%	—	—
C / 7.9%	*F* / 9.5%	$\bar{A}$ / 7.9%	*S* / 7.3%	—	—
D / 5.9%	D / 7.4%	*K* / 5.5%	***T*** / 4.1%	—	—
E / 3.8%	*E* / 3.2%	**L** / 4.1%	**U** / 3.8%	—	—
SOHO Traffic (GB)					
6.71	12.32	3.37	0.47	2.76	1.90
A / 42.4%	A / 46.2%	*M* / 27.4%	**V** / 16.5%	R / 14.0%	X / 71.7%
B / 11.3%	D / 12.0%	**N** / 8.3%	**W** / 13.4%	**Z** / 12.5%	S / 13.0%
D / 7.5%	B / 8.7%	***O*** / 8.3%	*X* / 9.3%	*H* / 11.1%	**BB** / 4.9%
G / 3.5%	C / 5.2%	**P** / 4.8%	**Y** / 7.1%	**AA** / 7.8%	**CC** / 1.6%
E / 3.2%	*H* / 2.7%	**Q** / 4.6%	*F* / 6.6%	$\bar{A}$ / 5.5%	**DD** / 1.4%

Table 3.10: The top 5 porn domains (sorted by bandwidth) across our traces. The top rows in Residential Traffic and SOHO Traffic represent the total bandwidth (GB) per trace. Domains with $\overline{\text{bar}}$ are blocked in the given trace. Underlined domains are blocked in the next trace. **Bold** domains are new domains, not seen in previous traces. *Italic* domains are unblocked in the next trace. Others are currently unblocked for which we do not have backward or forward reference.

this analysis to the same IP address and `User Agent` combination, which we assume to be stable over short time intervals. (This approach does not incorporate the possibility of multiple users behind a single IP address such as NAT employing the same user agent [142].) We then examine a histogram of domain names and search keywords[7] extracted from HTTP requests generated by the users. We make the following observations.

- On average 60% of the users perform a search engine query after encountering a block page for a porn domain, and 75% of users do so after encountering a block page for YouTube. Note that these proportions represent a lower bound because we lack visibility into encrypted traffic.[8] We find that for porn, content-specific searches heavily dominate the queries instead of searches for porn domains. This observation matches previous findings that porn users are flexible about served content as long

[7] We developed signatures to extract keywords from popular search engine queries.

[8] We find that among popular search engines, `google.com.pk` has a dominant presence in our data, also appearing in the top 5 servers in the SSL logs.

Trace	Total (GB)	Blocked domains (%)	Unblocked (%)
Residential Traffic			
21Dec11	2.67	9.00	91.00
28Feb12	2.32	3.94	96.06
SOHO Traffic			
21Dec11	3.37	0.16	99.84
28Feb12	0.47	0.29	99.71
18Sep12	2.76	10.70	89.30
02Aug13	1.90	0.01	99.99

Table 3.11: The distribution of porn bandwidth among blocked and unblocked domains. In 21Dec11 and 28Feb12, the censor only uses DNS for blocking; consequently, users can still access blocked content by using an alternative name server. In 18Sep12, although the censor uses IP blocking in addition to DNS blocking, the blocking is partial for some domains. In 02Aug13, the censor uses a combination of DNS blocking and HTTP redirection.

as it falls into a broad class [143]. For YouTube, we find a diverse range of primarily informational queries.[9].

- For porn, on average 70% of users who hit a block page access another porn domain within the next 5 minutes. For YouTube, on the day of the block, 7% of users view a video using an alternate video content provider, rising to 12% in 02Aug13. These figures run slightly lower than those from our survey where 17% of respondents indicated that they would use an alternate provider to access blocked YouTube videos. Tying this observation in with our earlier result, that showed that search queries are dominated by information-retrieval intent, we speculate that users primarily settle for non-video representations of information rather than actively searching for alternate or unblocked video providers.

- Surprisingly, we do not find a wide interest in either searching for circumvention mechanisms or directly accessing non-SSL Web proxies within our analysis time window. For porn, this might be the case because users have a tendency to shift to other unblocked porn providers, resulting in little incentive to try circumvention.

[9] Queries that represent user intent to obtain information about an object of interest, with potentially a large number of diverse results.

3.6 Effects on Content Providers

When censorship is imposed, users can respond in a range of ways: *(i)* stop accessing the censored content, *(ii)* access the same or similar content hosted by an alternate content provider, or *(iii)* employ a mechanism to bypass censorship and access the censored content. The first two options cause the censored content provider to lose a fraction of its previous traffic, and the second option may have a positive effect on the traffic for alternate content providers. The third option of employing circumvention technology potentially increases the cost of content distribution: the blocked providers have to serve content remotely because they cannot deploy servers in the censored region. Additionally, if the circumvention mechanism anonymizes user location, the censored content provider can no longer serve geographically relevant advertisements, which may reduce advertising revenue. This study does not concretely establish the economic implications of censorship on content providers; however, we highlight where this might be the case so as to stimulate future research.

3.6.1 Video Content

Table 3.8 illustrates the distribution of video bandwidth among the four major providers before and after the YouTube block. On average, YouTube provides about 97% of video content before the block.

On the day of YouTube block (18Sep12), only about 15% of video content is fetched from YouTube, half of which is being served by the ISP's cache servers and the other half is fetched from servers outside Pakistan. On this day, one of the two national service providers redirected HTTP requests for YouTube to one of its own error pages (Section 3.3). The residual YouTube traffic probably reflects the provider's insufficient capacity to handle the full traffic load; as a result, failing to consistently redirect traffic. Note that censorship was already underway when the traffic was captured because the error page initially appeared only five minutes into the trace. Moreover, we do not find any evidence of incomplete coverage in the blocking of YouTube's IP address space.

The trace collected 11 months after the YouTube block (02Aug13) does not contain any content served from YouTube. This does not necessarily imply that users stopped accessing YouTube: the fraction of encrypted traffic increased manyfold (see SOHO traffic in Table 3.7) from about 6% in 28Feb12 to over 30% of total traffic after the YouTube block in 18Sep12 and 02Aug13, indicating the use of SSL-based censorship bypass mechanisms (discussed in Section 3.5).

Table 3.8 shows that after the block, most of the video traffic generated from within Pakistan initially shifted to DailyMotion (82% of total traffic in 18Sep12), but 11 months later video traffic was split between DailyMotion (40.9%) and Tune.pk (57.6%). This traffic distribution is unusual considering the global traffic statistics of DailyMotion about a factor of 23 times more compared to Tune.pk [135]), indicating strong regional popularity.

Tune.pk is a Pakistani video portal that essentially provides a censorship-friendly wrapper around YouTube: it downloads YouTube videos and serves them from its own servers, providing users with the option to report offensive videos [144]. The case of Tune.pk highlights the benefits reaped by local markets as a result of the blocking of a competitor.

The overall shift in traffic potentially leads to a redistribution of advertisement revenue: the censored content provider loses out in favour of alternate unblocked providers. This trend is amplified when local video sharing websites link to videos from alternate providers that remain directly accessible to users. We find in Table 3.8 that a growing number of embedded links shift from YouTube to other video providers. Embedded links pointing to YouTube drop from an average of around 83% to 73% on the day of the YouTube block (18Sep12), and to about 51% 11 months later (02Aug13); DailyMotion gets approximately 32% of embedded links and Tune.pk jumps from virtually no embedding to nearly 11%.

The drop in the percentage of embedded YouTube links causes search engines to adjust their page ranks for localized searches. For example, a manual search (country-specific via `google.com.pk`) for top 5 local television shows reveals that the top results point to Tune.pk and DailyMotion, whereas top results for a search for non-local content (top 5 television shows in the US) point to YouTube.

In summary, the censored video content provider loses traffic and revenue to competing unblocked sites in multiple ways: direct reduction of traffic, local content providers move their hosted channels to alternate providers, reduced embedded referencing in third-party pages, and lower page rank for localized search. The provider potentially also loses revenue due to the increased cost of serving the content remotely to users who get around the block by employing circumvention technologies. For unblocked providers, these considerations may provide an incentive to take long-term control of local content. For example, DailyMotion moved to partner with the largest ISP in the country [145].

3.6.2 Porn Content

We ranked porn sites according to the corresponding traffic served into the country (measured using the methodology described in Section 3.4). Table 3.10 shows the top 5 (pseudonomized) porn domains for each trace. We observe that prior to the block, globally popular domains [135] top the list. After the porn block, new players emerge and take the top spots. In most cases, these new players are non-existent in previous traces (indicated in bold in the table). The relative distribution of the new domains varies inconsistently across the post-block traces. (We do see a few domains, such as X, S and R, that appear in the top 5 for more than one trace.) We speculate that the variety in the top ranked sites is caused by users, who used to be familiar with a small number of favourite porn websites, being spurred after the block to find out about alternatives through search engines. This explanation fits with the observation in Section 3.5 that after landing on a block page, porn users tend to perform content-specific search queries.

ASN (% of total video bandwidth)					
03Oct11	22Oct11	21Dec11	28Feb12	**18Sep12**	**02Aug13**
26.5 GB	56.5 GB	45.2 GB	12.6 GB	10.7 GB	2.7 GB
Local-ISP, PK (78.69)	Local-ISP, PK (82.08)	Local-ISP, PK (70.08)	Local-ISP, PK (76.21)	Dailymotion, FR (45.67)	FIBERRING, NL (58.67)
Google, US (17.85)	Google, US (13.74)	Google, US (24.74)	Google, US (17.62)	TMNET, MY (22.99)	Dailymotion, FR (19.76)
YouTube, IE (1.46)	YouTube, IE (1.68)	YouTube, IE (3.71)	YouTube, IE (3.15)	Local-ISP, PK (7.22)	OMANTEL, OM (14.01)
Dailymotion, FR (1.23)	Dailymotion, FR (1.41)	EdgeCast, US (0.68)	Dailymotion, FR (2.93)	Tinet, DE (4.11)	Akamai, US (7.70)
CCWW, GB (0.80)	Akamai, US (0.84)	Dailymotion, FR (0.62)	EdgeCast, US (0.12)	YouTube, IE (3.98)	Tinet, DE (0.68)

Table 3.12: The top 5 ASNs serving video, ranked by bandwidth. **Bold** indicates YouTube blocking. The top row shows the total video bandwidth. In our traces, FIBERRING serves Tune.pk videos, while OMANTEL, TMNET, CCWW, Tinet, Akamai, and EdgeCast primarily serve DailyMotion videos.

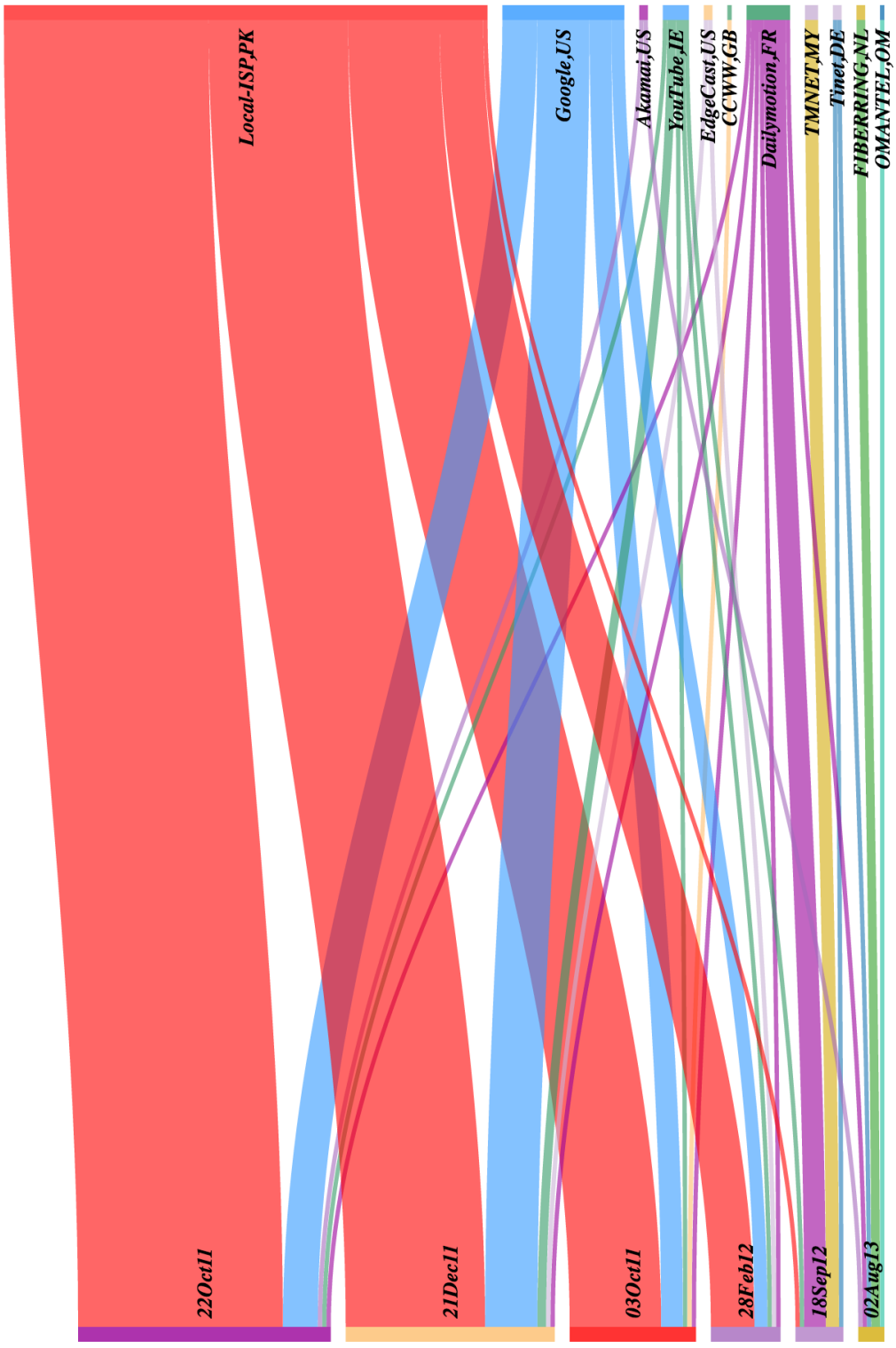


Figure 3.3: ASNs (right) that serve videos across our network traces (left). The link sizes correspond to bandwidth.

Censorship of a porn website impairs its revenue share from within the censored region. In Table 3.11, we analyze the distribution of bandwidth among blocked and unblocked porn sites. We observe that the bulk of the bandwidth is captured by unblocked porn domains. (An exception to the case is $\bar{A}$, which appears in a subsequent trace despite being blocked in 21Dec11—circumvention is made possible by obtaining correct IP addresses via non-local DNS resolvers.) The popularity of new porn domains is also explained by the fact that after the initial introduction of the porn blacklist (21Dec11), there seems to be no aggressive strategy by the censor to block new popular porn content.

3.7 Effects on Service Providers

In this section we assess the consequences of the censorship events on ISPs. We analyze the ISP's Web caching behaviour, with an emphasis on the video content downloaded from the four major providers because our pre-block traces indicate that videos constitute the majority of the content (about 95%) served by the ISP's cache servers.[10]

Table 3.12 lists the top 5 ASNs serving video content for each of the traces.[11] Prior to the YouTube block, the local ISP is the top ASN: on average, its caching servers provide 76% of the video content. On the day of the block (18Sep12), the ISP's caching servers provide a small fraction of YouTube video content. This leakage indicates that initially the ISP's censorship implementation was incomplete. The caching servers had not completely flushed cached YouTube content, continuing to serve it despite the block. (Recall that some users could potentially get correct answers for YouTube from their local DNS cache, or by using alternate DNS resolvers as discussed in Section 3.3.3.)

Moving forward 11 months, the ISP completely disappears from the figure to be replaced by CDNs serving DailyMotion and Tune.pk videos. Figure 3.3 illustrates the shift in video demand from the local ISP to CDNs in remote locations such as France, Germany, New Zealand, Malaysia, and Oman. The ISP's cache servers are completely absent from 02Aug13: the ISP confirmed to us that the systems no longer provided any utility. Based on discussions with the ISP operators, we learned two reasons for this. First, Google had provided free caching servers to Pakistani ISPs; infrastructure tailored specifically to cache YouTube. Other video content providers do not offer such caching solutions, making it difficult for the local ISPs to justify the cost of deploying and maintaining custom solutions for the providers' content. Second, the drastic decrease in unencrypted video content (Table 3.7) made it hard to justify the benefits of caching because ISPs cannot in general cache encrypted content. The ISPs instead turned to the option of leasing more upstream bandwidth, rather than buying and maintaining caching servers. Consequently, because

[10] We examine the distribution of `Content Type` served by the ISP's caching servers.

[11] Labovitz noted that after MegaUpload servers in North America were seized, content delivery shifted to European servers [125]. Our study investigates changes in infrastructural arrangements when the primary provider has been blocked, and the implications of such changes on ISPs.

all video content was primarily fetched from the servers of their respective providers, the ISP had to purchase additional Internet bandwidth after the block.

3.8 Summary

We studied the consequences of Internet censorship on service providers, content providers, and end users in the context of two censorship events in Pakistan, the blocking of porn content in 2011, and of YouTube in 2012. We analyzed home and SOHO traffic captured at a mid-size ISP in Pakistan before and after the censorship events. We developed methodologies to identify censorship activity within our network traces.

We observed that the porn block caused some increase in encrypted traffic, but primarily users turned to alternative unblocked porn sites. In contrast, the YouTube block caused a major shift towards encrypted traffic, indicating that users had resorted to circumvention mechanisms to access blocked content. In addition, we found this shift underway on the day when the government imposed censorship, suggesting that a portion of users rapidly adapted to the introduction of new blocking mechanisms.

Censorship of YouTube also affected the financial landscape of video content providers: new players emerged and completely took over the video-sharing market that was almost entirely owned by YouTube before it was blocked. This shift also had consequences for ISPs that used to serve video content mainly from YouTube caches hosted within their own networks. After the YouTube block, the ISP had to fetch video content through its upstream transit provider, leading an increase in bandwidth costs. After the YouTube block was locally enforced by the ISP via DNS redirection, there was a shift away from the use of the local ISP's DNS resolvers—dropping from more than 90% of all queries pre-block to about 70% post-block. Such a shift somewhat erodes a nation's overall control over its Internet traffic as users transfer their base of trust (i.e. DNS resolution) to parties outside the country.

Generality of Research Methodology and Findings. The methodology developed in Section 3.3 to detect DNS-, TCP/IP-, and HTTP-based censorship in historical network logs is generalizable and can be applied to other similar datasets to establish mechanism and targets of censorship. We restrict our analysis to YouTube and porn content in this study, but our methodology revealed the complete censorship blacklist.

We leverage the knowledge of censorship mechanism to quantify demand for blocked content before and after censorship events, and the consequences on various stakeholders. We are able to conduct this study because: *(i)* we have access to pre- and post-censorship data acquired at a suitable vantage point where traffic between ISP users and the Internet is visible, *(ii)* the censorship targets are significant enough to generate a measurable effect on various stakeholders, and *(iii)* the targets of censorship are already known through traditional media. We note that such favourable circumstances are typically

infeasible. However, if similar data is available, our methodology to quantify consequences of censorship may be applied with adjustments according to the targets of censorship.

In Section 3.4, we identify content providers that are potentially affected by censorship. For YouTube, we identify three other video steaming providers based on regional popularity derived from Alexa ranking. The same methodology can be used to identify key stakeholders where the target of censorship is a single well-defined service. For porn, which is a broad category of censorship potentially comprising hundreds of candidate websites, we turn to a commercial URL categorisation service offered by McAfee. This approach has two limitations: *(i)* our classification inherits the accuracy of McAfee's service, and *(ii)* the classification is conducted post hoc, implying that the porn providers in our data could have false positives (websites that did not serve porn at the time of data collection, but served porn at the time of URL classification), and false negatives (websites that served porn at the time of data collection, but did not serve porn at the time of URL classification).

We measure demand for content providers affected by censorship in terms of number of requests to these providers, and the downstream bandwidth, both identified by manually developed data-specific regular expression URI signatures. These signatures must be regenerated and evaluated for other datasets. Because of the large number of porn domains, manual signature generation is infeasible. We simply look for the presence of a domains classified as porn in the `Host` or the `Referer` part of an HTTP request—we lack coverage of porn requests or downloads where this is not the case. Moreover, our definition of demand for content is based on aggregate number of requests or bytes downloaded. This has the limitation that a small number of heavy hitters can inflate overall demand, incorrectly implying popularity. However, the advantage of this approach is that our measurements are not affected by multiple users behind a single IP (as is typically the case with NAT) as we consider the number of requests and bytes downloaded regardless of source IP addresses. We note that downstream bandwidth is a less generalizable metric to quantify demand as not all applications are bandwidth-heavy.

Demand is hard to quantify when a large fraction of traffic is encrypted, which is the case in the data captured after YouTube block (Section 3.6). We observe that demand for YouTube is redistributed to its competitors post-censorship, but this is accompanied by a significant increase in the fraction of encrypted traffic, and a significant decrease in the fraction of video traffic. It could well be the case that the demand for YouTube shifted to encrypted channel—so the competing content providers probably did gain some benefit from censorship, but the loss to YouTube is not striking.

Chapter 4

Differential Treatment of Anonymous Users

This chapter discusses an emerging class of Internet censorship where the user's connection arrives at the publisher unimpeded, but the publisher (or something working on its behalf) rejects it based on some characteristics of the source (e.g. the presence of 'undesirable' client-side software). Such blocking ranges from outright rejection to limiting users' access to a subset of the service's functionality or imposing hurdles like making users solve CAPTCHAs. To date, the observation of such practices has relied on anecdotal reports catalogued by frustrated users, for example Tor's "Don't Block Me" project maintains a user-reported catalog of services that cannot be accessed with Tor [146]. We methodically enumerate and characterize, in the context of Tor [147], the treatment of anonymous users as second-class Web citizens.

Anonymity networks serve an important purpose on the Internet, often providing the only means for citizens to access or distribute censored or restricted content without a threat to their privacy or even safety. A predominant example of such a network is Tor, the "king of high-secure, low-latency Internet anonymity" according to the NSA [148]. The growing trend of websites extending differential treatment to anonymous users undermines Tor's overall utility, adding a new dimension to the traditional threats to Tor (i.e. attacks on user privacy or governments blocking access to Tor). Section 4.1 provides background on Tor's design and the how it is typically blocked. In Section 4.2, we discuss prior work related with this study. We carry out a systematic characterization of websites and IP addresses that treat Tor users differently from normal connections, drawing upon several data sources: comparisons of Internet-wide port scans from Tor exit nodes versus from control hosts (Section 4.3), scans of the home pages of Alexa top 1K websites through every Tor exit (Section 4.4.1), and analysis of nearly a year of historic HTTP crawls from Tor network and control hosts (Section 4.4.2). We explore the techniques used by these websites, and how much of this differential treatment is due to explicit decisions to block Tor versus the consequence of fate-sharing; that is, when abuse generated from a Tor exit node triggers automated blocking mechanisms on websites, leading to *all* user

traffic generated from that exit node to be blocked. Section 4.5 concludes the chapter by discussing a number of potential methods that could minimize the impact of publisher-side blocking, and includes information about how to access data created during this study.

4.1 Background

For our anonymity system case study, we analyse Tor [47], the most widely used anonymous communication system, with over 2 million daily users [149]. Tor was designed to allow users to access TCP-based services (predominantly websites) privately and securely, preventing any intermediate agent from linking the user's identity to their activities. However, many Tor users primarily seek to circumvent censorship rather than to obtain privacy. Blocking access from Tor imposes serious limitations for Tor users, and has significant implications for Tor itself, potentially reducing its utility substantially. We provide a brief background on Tor's design and the different ways it is blocked.

4.1.1 Tor

Tor works by routing users' traffic over a three-hop 'circuit', with each hop being a volunteer-operated 'node' running the Tor software in server mode. Tor uses both per-link and end-to-end cryptography to provide confidentiality, integrity, and unlinkability between incoming and outgoing traffic at each hop. Tor users typically install the Tor Browser Bundle, which consists of a hardened Firefox-based browser and the Tor software configured as a client. When a user makes a request, the Tor client selects three nodes out of those available to form a circuit, connecting first to the 'entry guard', then through it to the 'middle node' and finally to the 'exit node'.

The exit node makes the TCP connection to the desired service and so is also the first target for abuse complaints from service operators. For this reason, not everyone is willing to operate an exit node, and the Tor server configuration allows operators to set an 'exit policy' stating to which IP addresses and ports the node will carry exit traffic. When a node activates (and periodically afterwards), it publishes a 'descriptor' to each of the 'directory authorities' which includes the IP address and port at which circuits can connect to the node, its exit policy, and its public key. The directory authorities together form and digitally sign the 'directory consensus', which they make available to clients both directly and via Tor nodes that act as 'directory mirrors'.

The directory consensus includes the information from each node's descriptor, but also includes a set of flags indicating in which positions a node can serve in the circuit (only sufficiently fast and stable nodes can serve as entry guards, and only nodes with a sufficiently permissive exit policy as exit nodes). Furthermore, the consensus includes a 'consensus weight' for each node, which is an integer proportional to the node's bandwidth capacity as measured by a set of 'bandwidth authorities'. When selecting a node for each

position in the circuit, clients first identify all the nodes that can take the respective position, and then select from these randomly, but biased by the consensus weights such that in aggregate they place a network load on Tor nodes in proportion to their capacity [58].

Because of its ability to circumvent censorship, the Tor network itself is subject to censorship. The simplest form consists of blocking access to the entry nodes by their IP addresses (which are easily found from the directory consensus). To counter this threat, Tor maintains a set of Tor nodes ('bridges') that act as entry points to the network but are not publicly listed in the consensus. Bridges are instead distributed to individuals in censored countries, making them harder to block reliably [147].

In reaction to this move, some countries fingerprint Tor traffic to block it, so Tor now allows the integration of 'pluggable transports' [150] that disguise the characteristics of Tor traffic. The use of bridges or pluggable transports does not affect how traffic exits the Tor network, so for the purpose of our study we do not deal with them specially.

4.1.2 Tor Blocking/Filtering

It is technically easy for Internet sites to block traffic from Tor relays on a wholesale basis, as there exist readily accessible and regularly updated lists of Tor relays. Internet services may have different reasons to apply such blocking: to discourage contributions by anonymous users, or avoid abuse such as comment spam. Inevitably, some well-meaning users will be excluded due to how widely Tor shares exit nodes across many users.

The first step to construct a Tor-specific blacklist is to collect the IP addresses of exit nodes. The easiest approach is to collect the IP addresses from the node descriptors in the directory consensus. However, these addresses denote the incoming IP address for nodes, and for nodes with multiple IP addresses this will not necessarily be the IP address for outgoing connections. As a result, using the IP addresses from the consensus could lead to both overblocking (by blocking the incoming IP address even though it is never used for outgoing exit traffic, but may have other uses) and underblocking (by failing to block the outgoing IP address because it is not an incoming address for any node). A more robust approach is 'active probing'—making Tor circuits that use each exit node in turn to establish a connection to a test server, and observing the originating IP address. This approach increases the accuracy of the list but puts more load on the network and reduces the frequency at which the list can be easily updated.

The second decision is which nodes to consider to be exits. The easiest option is to use the 'exit' flag assigned by the directory authorities if the node's exit policy permits at least two ports from 80 (HTTP), 443 (HTTPS) and 6667 (IRC). Relying on the exit flag results in overblocking because it is possible that an exit node will never be selected for a connection to a particular service using the blacklist even if it has the exit flag set (perhaps the service's IP address and/or port is excluded by the node's exit policy). Therefore, non-Tor users of the computer hosting the exit node will be blocked from

accessing the service even though there is no possibility that this computer will be the origin of Tor-originated abuse. There may also be underblocking if the node does not meet the criteria for the exit flag but its exit policy still permits connecting to the service in question.

Finally, the blacklist operator may decide to include some non-exits in the list (e.g. including nodes that have a 'deny all' exit policy and so can only be entry guards or middle nodes, or including IP addresses on the same netblock as Tor nodes). This approach is especially pernicious, as it leads to blocking of bystander IP addresses in ways that have little to do with Tor-sourced abuse. Motivations for doing so may include a desire to deter people from running Tor servers, or to mitigate underblocking that may occur as a result of missing Tor server configuration changes or mismatches between incoming and outgoing IP addresses for the node.

Examples of publicly available Tor blacklists include `dan.me.uk` [151], which optionally includes non-exit Tor nodes, and `dnsbl.sectoor.de` [152], which includes all IP addresses on the same /24 as the Tor exit by default. The Tor project itself maintains TorDNSEL [153], which uses active probing to increase accuracy, and also takes into account the specific service using the blacklist so as to reduce overblocking and underblocking.

To avoid complications resulting from these different approaches to blacklisting, we run our control probes from systems that did not share a /24 IP address with any Tor node, and our Tor-based probes from exit nodes that had the exit flag for at least a month, as well as permitting access to almost all IP addresses on port 80 (the destination port for our probes).

4.2 Related Work

We consider Internet censorship relevant to Tor from three perspectives: direct censorship of content, censorship of traffic *entering* Tor, and censorship of traffic *exiting* Tor. A large and growing body of literature focuses on the first two classes, but the latter category has seen little in the way of study; our work aims to fill this gap.

Much existing work has concentrated on measuring and evading direct content blocking in different countries, with an emphasis on state-level blocking of censorship circumvention systems. Dingledine *et al.* discuss when and how different governments tried to block access to Tor: governments mainly use address-based blocking of requests to the Tor website, relays, and bridges, and protocol-based blocking of TLS connections to the Tor network identified by Tor specific characteristics (e.g. cipher suite) [154].

Our work investigates a different aspect of the censorship problem. We examine *publisher-side* blocking of users; that is, blocking by the publishers based on the characteristics of the source, not blocking by an intermediate firewall based on characteristics of the destination. In the classical Internet censorship scenario, the publisher would be happy to accept connections from a client, but some network device near the client prohibits it.

Control nodes	
Number of control nodes	3
Number of IPv4 scans	7 per control node
Time span of scans	Aug 7–13, 2015
Scanned IP addresses per measurement	3,662,744,599
Average hit-rate per measurement	1.91% (σ=0.01%)
Tor exit nodes	
Number of exit node	4
Number of scans	4 per exit node
Time span of scans	Aug 10–13, 2015
Average hit-rate per measurement	1.87% (σ=0.03%)

Table 4.1: Summary of control and exit node data. For all scans, we filter out IP addresses included in the largest blacklist (i.e. one employed by the last scan). Network loss per measurement is estimated as the percentage of IP addresses inaccessible from a node but accessible from at least one other node.

We, on the other hand, look at cases where the client's connection arrives at the publisher unimpeded, but the publisher (or something working on its behalf) rejects it.

In our work we make use of data from the Open Observatory of Network Interference (OONI) [155]. (Despite a similarity in purpose and acronym, this project is separate from the OpenNet Initiative discussed later.) The OONI dataset has a crucial feature for studying differential treatment of Tor users: it consists of many simultaneous downloads both with Tor and without Tor. While the intent behind these measurements is to highlight content that is inaccessible from certain locations *unless* one uses Tor, we can employ the same information to identify destinations inaccessible *because* one uses Tor.

Developing robust techniques to detect blocking is also important. We need to know when an application is being blocked, and we also need to distinguish genuine network interference from benign or transient failures. Jones *et al.* tested automated means of detecting censorship block pages in an OpenNet corpus [156]. A metric based on page length proved the best-performing of several options. Our experiments necessitate different ways of detecting blocks at different network layers. In Section 4.3, we use repeated scans across space and time; and in Section 4.4, we compare test downloads against simultaneous control downloads.

4.3 Measuring Network Layer Discrimination

As we discuss in Section 4.1, a straightforward technique for services to block Tor is to filter traffic from publicly listed exit nodes. To broadly assess this, we measure Tor filtering using ZMap probing from both Tor exit nodes and from control (non-Tor) nodes to see how their access to remote addresses differs. For convenience we term these measurements as assessing 'network layer' discrimination, even though from a technical perspective these combine measurement of layer-3 and layer-4 blocking, since we restrict our measurements to attempts to connect to TCP port 80 services.

4.3.1 ZMap

ZMap is a high-performance network scanner capable of scanning the entire IPv4 address space in as little as 45 minutes, much faster than traditional scanners such as Nmap [157]. ZMap achieves this efficiency by incorporating multiple optimizations such as randomized target selection and maintaining no connection state. Because ZMap does not maintain state, it also does not retransmit probes in case of loss. We used ZMap for test runs of the entire IPv4 address space starting in Spring, 2015. Over the course of repeated experiments, we uncovered several bugs (for some of which we contributed fixes, while others were fixed by the ZMap team), addressed measurement considerations (for avoiding measurement loss), and added extra functionality as discussed later. For our measurements, we recorded both TCP SYN-ACK and RST packets. We configured ZMap to run at 100 Mbps rather than at 1 Gbps to avoid saturating our local networks. Doing so results in one scan taking about 7 hours rather than 45 minutes.

4.3.2 Overview of Measurements and Block Detection

We run our scans from Tor exit nodes and from control nodes based in three universities. We compare responses to our Tor scans with those from the baseline control scans and flag deviations as potentially reflecting discriminatory blocking. Target hosts respond to ZMap probes (TCP SYNs) in one of three ways: *(i)* sending a SYN-ACK, which we term a *successful* response, *(ii)* sending a RST, which we term an *unsuccessful* response, or *(iii)* not responding, which we also deem an unsuccessful response. ZMap, by default, only records successful responses; we modified it to record RSTs as well. We note that for an individual probe, it is not possible to distinguish a lack of response from packet loss.

We might in simple terms think that we can identify Tor blocking by observing destination addresses that respond to probes from the control nodes but not to those from the Tor exit nodes. However, this reasoning has two main limitations: *(i)* unsuccessful responses could arise because of *packet loss* along either the packet forward or return path, and *(ii)* destinations can respond inconsistently to probes because of factors unrelated to discriminatory blocking such as servers only operating during certain hours of the day or

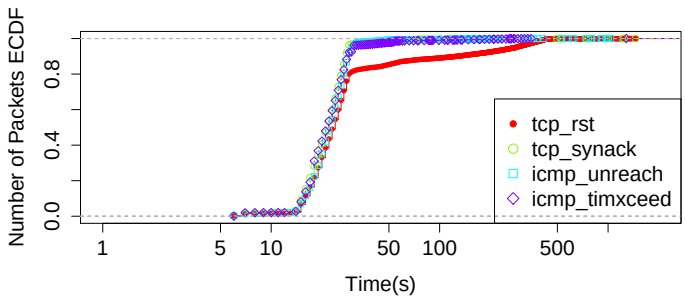


Figure 4.1: The distribution of time until receiving a response packet since the last probe was sent, for a full scan of IPv4.

days of the week.

More generally, we need to consider issues of *churn*: how Internet service reachability varies, in both spatial and temporal terms. By *spatial churn*, we mean the notion that simultaneous probes sent from topologically separate clients to the same server might yield different outcomes; for example, due to network congestion or a network outage blocking the path from one of the clients but not the other. By *temporal churn*, we refer to the reachability from the same client to the same server varying over the course of time; for example, due to day-of-the-week effects governing when the server is accessible.

Thus, to understand how to soundly compare probe outcomes seen at our control nodes versus from Tor nodes, we need to incorporate consideration of how to distinguish probing results that differ due to churn versus those that actually reflect discrimination. Note that through out the rest of our discussion, the underlying assumption is that services either completely block a Tor exit node or allow it: we do not deal with selective blocking or rate-limiting in this study.

4.3.3 Mitigating the Effects of Packet Loss

As noted above, ZMap does not allow us to distinguish between a single non-response and a packet loss event. To account for this limitation, we take care to minimize loss in our measurements and to account for potential packet loss in the network.

Mitigating Measurement Loss

We first ask whether ZMap accurately sends all the packets it is configured to send, and whether it correctly logs packets and responses.

We profiled ZMap using an experimental setup that consists of a well-provisioned machine running ZMap, and a separate machine running a packet capture. All the ZMap packets are directed to the second machine via a Gigabit Ethernet cable. Separating packet transmission and packet capture allows us to account for losses occurring because of both ZMap itself and the underlying network card. It also avoids the scenario where the

two processes compete with each other for CPU cycles. When ZMap runs with its default configuration, we see a 6.7% failure rate—this failure is completely eliminated when we throttle our sending rate down from 1 Gbps to 100 Mbps. (During this process, we also identified and reported a bug in ZMap that caused it to not send certain packets due to the interaction between scan targets, the blacklist, and thread-level sharding.)

In addition, we need to configure a timeout for ZMap to deem that a packet did not receive a response. Figure 4.1 shows the distribution of the time measured between sending the last scan packet and receiving a response for a full scan of IPv4. To generate this plot, ZMap logged response packets for 25 minutes after sending the last scan packet. More than 95% of all replies (excluding RSTs), and 80% of RSTs arrive within the first 30 seconds, while the rest trickle in up until 500 seconds. Although unusual, late responses could arise due to backed-off timers in the case of SYN-ACKs, huge bufferbloat, or initial latency incurred by extensive setup requirements of cellular wireless devices [158]. Given this data, we choose a conservative cooldown value of 10 minutes for responses to come in.

Network Packet Loss

An unsuccessful response can be due to loss on the paths between the scanner and the destination, caused by transient network issues such as congestion or network failure. We reduce such noise by sending redundant probes per destination. If any of the probes elicits a SYN-ACK from the destination, we treat it as a successful response, because a single response suffices to inform us that the target server does not block Tor traffic.

We can introduce probe redundancy in many ways; the simplest is by conducting back-to-back scans from the same vantage point. However, since a single scan takes about 7 hours to complete, such an approach introduces a large gap between the redundant probes, which can lead to inconsistent responses due to temporal churn. We ran 3 back-to-back scans from one of our control vantage points. We observed a temporal churn between the first two scans of 13.30%, which increased to 21.61% when computed across the three scans. We repeated the experiment at another of our control vantage points and made similar observations. This finding means that servers respond quite inconsistently across long intervals of time.

This high temporal churn motivates us to incorporate redundancy at shorter timescales in our measurements. Although ZMap allows us to send multiple probes per target in a single scan, it does so back-to-back without any delay between them. This approach only helps if loss events are independent; however, transient network issues mean that loss events are presumably not independent.

Since ZMap does not keep state, we cannot retransmit only those probes for which we did not receive a response. We therefore follow the simple strategy of sending K probes, resending them, sending another K probes, resending them, and so on. For $K = 1,000,000$ and with a sending rate of 100 Mbps, this means that the retransmitted probe follows 6.7 seconds after the original. This approach allows us to maintain the sending bandwidth

Exit Node	Location	Uptime	Bandwidth (MB/s)
Axigy1	USA	35 days	31.09
Axigy2	USA	76 days	31.46
NForce2	Netherlands	35 days	31.46
Voxility1	Romania	1 day 17 hr	16.99

Table 4.2: Description of Tor exit nodes from which IPv4 scans are conducted.

Footprint	IP Addresses	Axigy1 (%)		Axigy2 (%)		NForce2 (%)		Voxility1 (%)	
		orig.	ret.	orig.	ret.	orig.	ret.	orig.	ret.
Raw	103,329,073 (2.82%)	16.05	15.48	15.45	14.01	17.66	16.18	16.20	14.65
Lax	99,547,512 (2.72%)	14.09	13.50	13.68	12.19	16.14	14.59	14.63	13.01
Strict	52,148,437 (1.42%)	1.91	1.91	1.25	1.23	2.59	2.55	1.88	1.82

Table 4.3: Blocking of Tor exit nodes across the Web footprint. We show the footprint as % of probed IP addresses (3,662,744,599). For each exit node, we present the original (*orig.*) block proportion of the footprint and that retained (*ret.*) after weeding out false positives using 5 verification scans.

and allows us to keep ZMap as a single threaded process; however, it doubles the length of a full scan (as expected). Across three sites and four scans, we found that factoring in responses to retransmitted probes increases the response rate for original probes by 1.04% (we can distinguish these by sending retransmissions from different ports). We further observe that there is a temporal churn of 1.93% between 6.7 seconds apart scans, which is significantly lower than 13.30% churn for scans run back-to-back (effectively about 7 hours apart).

4.3.4 Data

We run measurements from a set of three control nodes and a set of four Tor exit nodes. Two control nodes are located in US universities (University of California, Berkeley, and University of Michigan) and one in a European university (University of Cambridge). The control node measurements allow us to calibrate and understand our measurement method and the data, and provide the baseline measurements against which we compare the Tor exit node measurements.

Our first goal is to develop a global 'Web footprint', a set of IP addresses that respond to our scans on port 80. On average, a control node sees a hit rate of 1.91% (σ=0.01%) per measurement scan (translating to about 70 million IP addresses). We note that each scan consists of two probes per target IP address (Section 4.3.3); a 'hit' consists of a

SYN-ACK response to our SYN for at least one probe. This number is roughly constant across the three locations. However, no two scans return the same set of IP addresses; reasons include routing and transient failures, network policies, time-of-day effects, and regular usage patterns (the issue of *churn* discussed previously).

We first conducted extensive preliminary ZMap scans (on the order of 90 scans over a period of 3 months) in order to calibrate the accuracy of our measurement methodology and address problems that arose. All the scans employed a blacklist excluding IP addresses, which we added to whenever we received a request. We run Web servers on the control nodes that identify our scanning activity as research and provide an email address for sites to opt out of further scanning.

During our measurements from March to August, 2015, we received scan exclusion requests from a total of 134 unique email addresses for 426 networks (covering a total of 3,532,751 hosts). Note that this number provides an upper bound as the machines at Michigan and Berkeley use site-wide scan notices, implying that a complaint could have been triggered by any of the scans running from these sites.

Once fully developed and debugged, for our final analysis we gathered 37 full IPv4 scans over a period of 7 days, conducting 16 from four Tor exit nodes. Table 4.1 shows the breakdown of the measurements run from the control and Tor exit nodes. We now turn to analyzing the final data to understand temporal churn (how the footprint changes across scans spanning multiple days) and spatial churn (how our view of the global Web footprint set changes across the three control locations).

Temporal Churn. For the same location, we see significant differences in the number of IP addresses that successfully respond, even between consecutive days, ranging up to 17%. Figure 4.2 shows the number of new IP addresses that each site successfully contacts per day. Using the first day as the baseline, this value gradually drops from a peak of about 7 million on the second day to about 4 million on day 7. The slow convergence rate indicates that temporal churn remains high even for the same location, and that obtaining a true underlying Web footprint for a given location may not be well-defined. Temporal churn is likely caused by nodes that only come online occasionally; however, we do not investigate the reasons in this study.

Spatial Churn. Not all IP addresses respond to all three control locations, even though we initiated the control scans all at the same time for each run. One potential cause for this phenomenon is wide-area routing issues. We identify IP addresses that only successfully responded to one or two locations (not all three) as reflecting spatial churn, corresponding to about 3.66% (about 3.7 million) of responding IP addresses across the footprints from the three control nodes. Upon further investigation, we observed that 52% of this spatial churn arose from IP addresses accessible from only *one* of the control nodes. We tested a handful of these IP addresses manually and confirmed this behaviour, ruling out that it

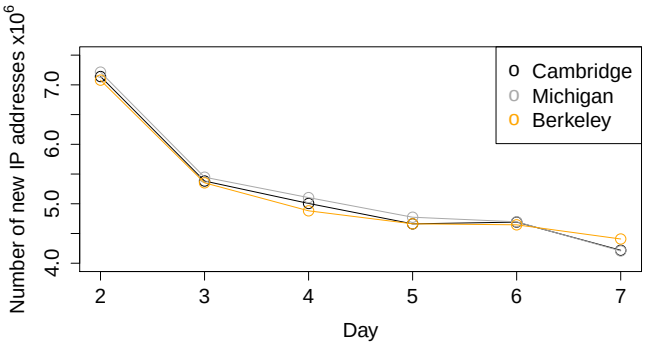


Figure 4.2: The number of new IP addresses each control node sees per day.

reflected a ZMap problem.

Given the significant amount of spatial and temporal churn, we settle on two definitions of Web footprint to use for our analysis: *(i)* a LAX definition, where we only remove cases of spatial churn, considering the set of IP addresses for which all control nodes see a response at least once across the seven days, and *(ii)* a STRICT definition, where we remove cases of both spatial and temporal churn, including in this set only IP addresses for which all control nodes receive a successful response on all days. We find that the RAW footprint contains 103,329,073 IP addresses (2.82% of the probed set). The LAX footprint is 96% of the RAW footprint, whereas STRICT reduces the RAW footprint to 50%.

Ethical Standards. The key ethical considerations here are that the scans should not undermine security of the targets, and that the scans should not disrupt or unreasonably degrade the functionality of key stakeholders—the universities where our control nodes are located, Tor exit nodes, and the scan targets. We sought permission from university security teams, clearly describing the nature, duration, frequency, and bandwidth of our scans. The teams forwarded any scan-related complaints to us. To allow autonomy of targets with respect to participation in the experiment, we ran a Web page on scan machines to explain our experiment and to provide the option to be excluded from the experiment. Whether received from university technical staff, or directly through our scan notice page, we processed all complaints within 24 hours. We adopted a similar procedure for Tor exit nodes. Unlike the control nodes, the exit nodes were not dedicated scan machines and resources had to be shared with multiple Tor instances per machine. To avoid congesting the upstream link, and to minimize load on the network interface, we turned off all but one Tor process on these machines. Consequently, during the 4 days of the experiment, the exit nodes handled less Tor traffic than their original capacity—potentially leading to increased burden on other exit nodes. We believe that the effect is minor because there are about 900 other exit nodes, enough to handle the additional bandwidth. Moreover, any inconvenience was caused for only 4 days, which is outweighed by the benefits offered by this study in illuminating network-layer discrimination of Tor.

Our scans comprise 2 probes per target in 24 hours period, with a gap of about 6

seconds between consecutive probes. This traffic is too small to cause any performance degradation to the target. A more serious concern is that 2 probes per target sent to an entire network might lead to increased load on network devices. ZMap avoids congesting target networks by sending probes to a random permutation of IP addresses instead of sequentially scanning the IP address space. The probe itself comprises a TCP SYN on port 80, which does not pose any security risk to the target.

4.3.5 Assessing Network Layer Discrimination

We conducted the scans from four high-bandwidth Tor exit nodes over 4 days (August 10–13, 2015) (Table 4.2). These represent 3% of aggregate Tor exit bandwidth. We note that each exit node hosts 2–3 Tor processes on the same interface. As our 100 Mbps scans use the same IP address as the Tor exit node, we turned off all but one Tor process on these machines for the duration of the experiment to minimize load on the interface and potential packet loss on the interface or the outgoing link. These preventive measures helped reduce our reported `pcap` loss on the exit nodes to 0.001% of the typical number of responses seen per scan. We also chose Tor instances that use the same IP address for incoming and outgoing Tor traffic to allow our scans to trigger even 'lazy' blacklists.[1] For three of the exit nodes, we displayed our scan notice page on port 8080 instead of the usual port 80, as the latter already displayed a separate Tor abuse complaint page.

Our basic technique for flagging network layer discrimination of Tor is to identify the part of the Web footprint that *never* produces a successful response to a Tor exit node. We examine this separately for each exit node, as we do not assume that all the exits are blocked consistently. Once we have extracted this subset for an exit node, we scan the suspicious IP addresses 5 times from the corresponding exit node and discard IP addresses that respond successfully at least once, effectively reducing our false positives. As a result of the last step, the blocked IPs per exit node reduce on average by 7.70% (σ=2.82%) for Raw footprint, 8.94% (σ=3.23%) for Lax footprint, and 1.05% (σ=0.74%) for Strict footprint. We note that our approach does not account for transient IP layer blocking such as abuse-based filtering. However, assuming that transient IP blocking is enforced for a time window smaller than 4 days, we may still observe a successful response in scans conducted before or after the transient block. Using this methodology, we characterize Tor blocking for both Lax and Strict Web footprints.

[1] The easiest approach to blacklist Tor is to block IP addresses from the node descriptors in the directory consensus that denote the incoming IP address for nodes. This blacklisting approach fails to cover nodes that use a different IP address for outgoing traffic, as discussed in Section 4.1.2.

Axigy1 (13.50%)	Axigy2 (12.19%)	NForce2 (14.59%)	Voxility1 (13.01%)
CHINA169-BACKBONE, CN (11.33)	CHINA169-BACKBONE, CN (11.73)	CHINA169-BACKBONE, CN (11.02)	CHINA169-BACKBONE, CN (12.53)
CHINANET-BACKBONE, CN (7.42)	CHINANET-BACKBONE, CN (8.20)	CHINANET-BACKBONE, CN (7.30)	CHINANET-BACKBONE, CN (7.93)
Uninet S.A., MX (3.43)	DCI-AS(ITC), IR (3.26)	AIRTELBROADBAND-AS-AP, IN (4.31)	DCI-AS(ITC), IR
DCI-AS(ITC), IR (2.94)	Uninet S.A., MX (3.00)	BSNL-NIB, IN (4.30)	Uninet S.A., MX
BSNL-NIB, IN (2.94)	DTAG Deutsche Telekom, DE (2.89)	DCI-AS(ITC), IR (2.73)	DTAG Deutsche Telekom, DE

(a) Lax Web Footprint (99,547,512 IP addresses forming 2.72% of probed IPv4)

Axigy1 (1.91%)	Axigy2 (1.23%)	NForce2 (2.55%)	Voxility1 (1.82%)
MCCI-AS, IR (11.91)	MCCI-AS, IR (18.44)	MCCI-AS, IR (8.92)	OCN NTT, JP (20.90)
RMH-14-Rackspace, US (10.87)	DREAMHOST-AS, US (13.07)	RMH-14-Rackspace, US (8.14)	MCCI-AS, IR (12.46)
RACKSPACE-Rackspace, US (9.92)	KUNET-AS, KR (3.59)	RACKSPACE-Rackspace, US (7.43)	DREAMHOST-AS, US (8.83)
DREAMHOST-AS, US (8.44)	REDSTATION, GB (2.66)	DREAMHOST-AS, US (6.32)	GO-DADDY-COM-LLC, US (3.51)
Rackspace Ltd., GB (5.85)	SINGLEHOP-INC, US (2.09)	BBIL-AP BHARTI Airtel, IN (5.59)	FBDC FreeBit, JP (2.56)

(b) Strict Web Footprint (52,148,437 IP addresses forming 1.42% of probed IPv4)

Table 4.4: The ASN distribution (top 5) of IP addresses in Lax and Strict footprints that block Tor across the four exit nodes. For each exit node, we show the percentage of the footprint that blocks it, and the ASN distribution (%) of the blocking IP addresses in the footprint.

Axigy1	Axigy2	NForce2	Voxility1
MCCI-AS, IR	DREAMHOST-AS, US	RMH-14-Rackspace, US	OCN NTT Communications, JP
RMH-14 - Rackspace, US	KUNET-AS, KR	RACKSPACE - Rackspace, US	DREAMHOST-AS, LLC, US
RACKSPACE - Rackspace, US	REDSTATION, GB	AIRCEL-IN Aircel Ltd., IN	KUNET-AS, KR
DREAMHOST-AS, LLC, US	LLC-SK-CONTINENT, RU	DREAMHOST-AS, LLC, US	BEKKOAME INTERNET INC., JP
CNNIC-SGATHER-AP, CN	tropicalweb-as, MZ	Rackspace Ltd., GB	tropicalweb-as, MZ

(a) Lax Web Footprint

Axigy1	Axigy2	NForce2	Voxility1
MCCI-AS, IR	MCCI-AS, IR	MCCI-AS, IR	OCN NTT Communications, JP
RMH-14 - Rackspace, US	DREAMHOST-AS, US	RMH-14 - Rackspace, US	MCCI-AS, IR
RACKSPACE - Rackspace, US	KUNET-AS, KR	RACKSPACE - Rackspace, US	DREAMHOST-AS, US
DREAMHOST-AS, US	REDSTATION, GB	DREAMHOST-AS, US	KUNET-AS, KR
Rackspace Ltd., GB	AS-INTERMEDIA, US	Rackspace Ltd., GB	BEKKOAME INTERNET INC., JP

(b) Strict Web Footprint

Table 4.5: The ASN distribution (top 5) of IP addresses (by fraction in their subnet) in the Lax and Strict footprints that block Tor exit nodes. As multiple ASNs perform 100% blocking of Tor, we further order them by ASN size (the number of IP addresses in an ASN).

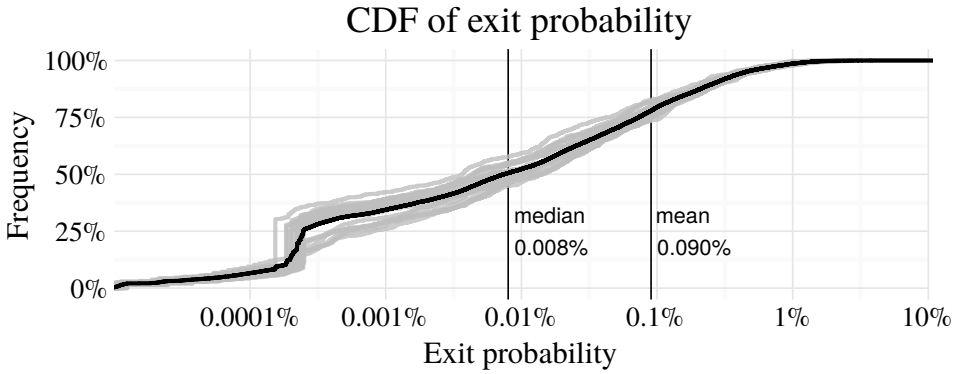


Figure 4.3: The cumulative distribution of exit probability from 20 randomly sampled Tor consensuses since September, 2014. The bulk of exits have between 0.001% and 1% probability of being selected. The largest exits tend to have an exit probability around 5%. The notable rise just above 0.0001% is an artefact of the bandwidth measurement process; when a node's bandwidth is unmeasured for whatever reason, it receives a default bandwidth of 20 KB/s, giving it a very low exit probability of around 1 in 500,000.

Table 4.3 shows the breakdown of the Tor blocking we detect. We detect a significantly higher rate of blocking for the Lax footprint compared to Strict (13.01–16.14% and 1.23–2.59%, respectively). This discrepancy could be caused by multiple factors. First, the Lax footprint is more than double the Strict footprint, due to the weaker selection criteria. This means that it is likely to see more churn and therefore has a larger potential for false positives. Second, as we see next, the Lax footprint exposes large access ISP networks, which potentially block Tor across the whole network. Due to the transient nature of nodes in such networks, these are less likely to be seen in the Strict footprint.

Tables 4.4 and 4.5 show the breakdown of the ASNs that block Tor. Tables 4.4 shows the distribution by the number of IP addresses in an ASN that block Tor, for both the Lax and the Strict footprints. We see that the ASNs in the Strict footprint are dominated by hosting services, which suggests that the blocking could be policy or abuse-driven. The Lax footprint contains ASNs that are potentially access and mobile ISPs, such as CHINANET, BSNL, and Airtel. These ISPs likely enforce symmetric blocking of Tor. Because these are access ISPs, nodes in such networks are more likely to go offline, which explains their absence in the Strict list. We note that ASes of IPs in the Lax footprint that block Tor traffic mostly are based in countries that are well-known for their censorship practices, such as China and Iran. So far these countries have been reported to block *access* to Tor network, but our results suggest that traffic coming *from* Tor network may also be blocked either as a policy or as an unintended effect of the mechanism of censorship chosen.

Table 4.5 shows a similar result sorted by the proportion of servers within a given ASN that block Tor. We see a higher prevalence of hosting sites in both Lax and Strict.

4.4 Application Layer Discrimination

We have seen that Tor exit nodes encounter a restricted Internet at layers 3/4. In this section, we describe our experiments to measure discrimination of Tor users at layer 7. We base our observations on two data sources: *(i)* five days of our own intensive scans of 1,000 URLs from a control server and through every Tor exit node, and *(ii)* a year's worth of paired Tor–non-Tor scans of over 2,300 URLs from the Open Observatory of Network Interference (OONI). OONI is a global network measurement platform aimed at detecting censorship and surveillance; one of its tests is particularly suited to our study.

There are two ways in which Tor users may find themselves blocked by a server. The server may specifically block Tor users using a blacklist of Tor exit node addresses: the only maintenance required is keeping the blacklist up to date. Alternatively, Tor users may simply be caught up in an automated blocking system that does not target Tor in itself, but merely reacts to the consolidated traffic of multiple users that come from an exit node. Perhaps the most conspicuous current example of this phenomenon is CloudFlare's "Attention Required!" CAPTCHA page. CloudFlare is a large content delivery network (CDN) that by default assesses the 'reputation' of each client IP address in terms of how much malicious traffic it has been observed to send, and blocks attempted access by clients with sufficiently poor reputations. A CloudFlare support page explains that while they do not specifically target Tor users, "due to the behaviour of some individuals using the Tor network (spammers, distributors of malware, attackers, etc.), the IP addresses of Tor exit nodes generally earn a bad reputation" [159].

Some sites—mainly larger Web properties, though not exclusively—apparently implement their own detection algorithms and custom block pages. Notable examples in this category are Craigslist and Yelp. Many other sites simply inherit the blocking behaviour of their Web host or content delivery network, which may or may not offer customers control over the severity of Tor blocking. In this latter case, a single provider's policy can affect many websites.

Our first experiment—contemporary scans of Alexa URLs—provides broad coverage across all Tor exit nodes over a short time period. The second experiment—analysis of historic OONI scan data—covers a long time span and more URLs, but lacks a longitudinal comparison across all exits for each URL. Figure 4.3 illustrates the rationale for conducting our own scans in addition to analyzing past data. Tor clients do not choose exit nodes with equal probability; each exit is weighted according to its bandwidth [58, §3.8.3]. Faster nodes have a greater probability of being used (subject to some other constraints such as exit policies). The OONI data reflects Tor circuits made in the ordinary fashion; therefore, low-probability exits are rarely represented. Measuring low-probability exits is important because it helps to distinguish the two kinds of blocking; slow exits will appear on blacklists but will have fewer users and thus be less likely to exceed abuse thresholds.

Our measurements are limited to home pages, except for about 3% of OONI URLs that include a path component. The Alexa URLs are only home pages. We know through

experience that some application layer blocking only becomes apparent when accessing certain deeper features or pages of websites. For example, Wikipedia allows Tor users to read articles but they cannot edit articles [160], Google allows access to its home page but may present a CAPTCHA or block page when doing a search, and Bank of America does not permit Tor users to log in. We did not explore such deeper blocks, which would require extensive additional methodology to study in a large-scale fashion.

4.4.1 Contemporary Scans

To measure the differential treatment of Tor users, we visit Alexa top 1K URLs once from all available Tor exit nodes and once without Tor. For the former, we use Exitmap [161], a fast and extensible Python-based scanner for Tor exit nodes. Exitmap uses Stem [162] to connect to the Tor network, and enables running a module over all available exit nodes. It is designed to monitor the reliability and trustworthiness of exit nodes [163] but its basic architecture is generic and it can be used to run any query.

Exitmap downloads a Tor consensus and extracts the currently available exit nodes. It then initiates circuits using the selected exit nodes as their last hop. To improve the performance of the scanning process, Exitmap uses two-hop circuits instead of the default three-hop circuits. (Using single-hop circuits is not an option because by default, exit nodes do not allow direct connection from other non-Tor IP addresses [164]; additionally the authors of Exitmap argue that one-hop circuits may permit exit node operators to treat scanning connections differently [163, §3.2].)

To measure discrimination against Tor, we send HTTP requests per URL with Tor through every available exit node, and one HTTP request per URL without Tor. We use Exitmap to build Tor circuits and a Python program to send the HTTP requests. Our experiment of downloading thousands of URLs per exit node stretched Exitmap past its original design parameters, requiring us to overcome some scanning challenges. Downloading a single URL using all the Tor exits requires 45–50 minutes on average; however, much of this time consists of the circuit-construction overhead. We reduced the total scanning time by running 5 instances of Exitmap in parallel, and downloading 20 URLs at a time through each circuit. With these changes, visiting 100 URLs through every Tor exit node takes around 1–2 hours on average. By default, Tor rebuilds circuits and streams each hour: we set configuration parameters to prevent this

We collect data over 5 days, from August 10–14, 2015. We select exit nodes that allow traffic through port 80 and 443. Different runs of Exitmap can select different exit nodes for two reasons. First, the Tor directory authorities release a fresh consensus, listing available nodes, every hour. New nodes might appear, old nodes might disappear, and nodes can change their ‘exit policy’ of allowed ports. Consequently, the available exit nodes can change every hour. Second, to build circuits, Tor clients need to download ‘enough’ of the network so that they can construct a sufficiently large number of the possible paths through the network.

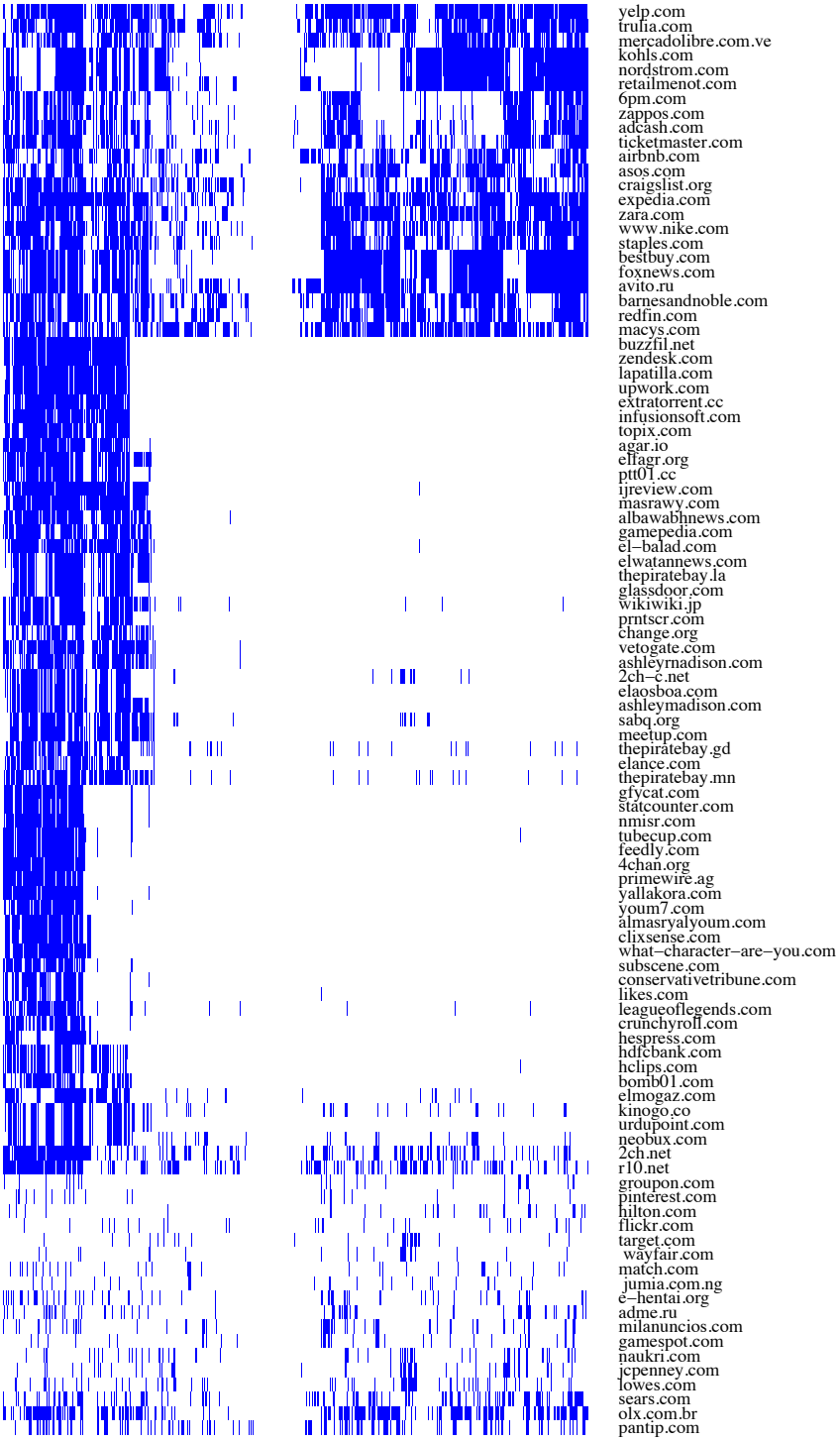


Figure 4.4: The top 100 websites among the Alexa top 1K that block most Tor exit nodes. Each row corresponds to one website. Each column represents one exit node. A blue bar represents a blocking event; that is, the Web server responded with a `200` status code when accessed without Tor and another valid but non-`200`-level HTTP response when accessed with Tor. No site blocked all exit nodes. During our scan, on average 15 sites blocked over 50% of the exit nodes; `yelp.com`, at the top of the figure, is one such site.

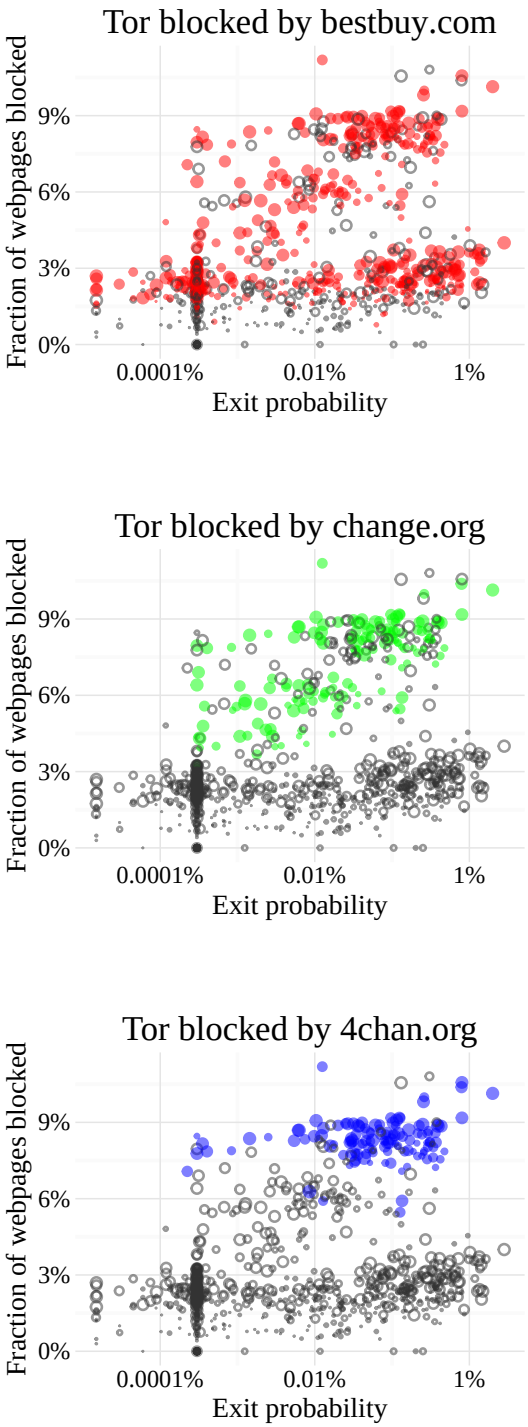


Figure 4.5: The relationship between blocking rate and exit probability. Each dot represents one exit node. The size of a dot represents the age of an exit (i.e. how long has it been since the node became an exit) in days. Bigger nodes are older than the smaller nodes. The colour of the dots represents different sites. The topmost graph shows the number of exit nodes blocked by `bestbuy.com`, which is hosted on Akamai. `bestbuy.com` and other Akamai sites block over 60% of the exit nodes. The middle graph shows the number of exit nodes blocked by a site hosted on CloudFlare, `change.org`. The last graph shows the number of exit nodes blocked by another site hosted on CloudFlare `4chan.org`. `4chan.org` uses a less strict setting than `change.org`, and thus blocks mostly high probability and older exit nodes.

Date	# Exits	Request		Response (200)		Response (non-200)		Non-HTTP Errors	
		Control	Per Exit	Control	Per Exit	Control	Per Exit	Control	Per Exit
Aug 10, 2015	908	1000	741.29	992	641.93	1	26.03	7	73.33
Aug 11, 2015	915	1000	679.43	985	595.21	3	23.93	12	60.29
Aug 12, 2015	905	1000	735.66	986	632.96	6	28.22	8	74.48
Aug 13, 2015	915	1000	735.46	989	639.44	4	26.84	7	69.18
Aug 14, 2015	899	1000	738.22	989	641.55	2	28.18	9	68.49
Average	908	1000	726.01	98.82%	86.81%	0.32%	3.67%	0.86%	9.53%

Table 4.6: The total number of HTTP requests sent and the responses received during the 5 days of scanning. On average, over 3.67% of the Alexa top websites discriminate against Tor (p-value of the permutation test = 0.008).

By default, Tor uses a value chosen by the directory authorities, which can change every hour. Thus, different exit nodes might be selected at different times. During our data collection period, we collect data from 899–915 exit nodes. The distribution of the exit probabilities in our dataset is within the regular range as seen in Figure 4.3. We have a good mixture of both high- and low-probability exit nodes. We find that 83–89% of the circuits succeed. This success rate is similar to previous exit scanning studies [163].

Table 4.6 shows the total number of HTTP requests sent and responses received on each day. We send 1,000 HTTP requests through over 900 exit nodes, but the average number of requests per exit node is less than 800. This discrepancy occurs for two reasons. First, as mentioned in the previous section, different exit nodes can be selected during different runs of Exitmap because the available exit nodes can change every hour. As our crawl takes over 10 hours to finish, some of the exit nodes are not available for the whole span of crawling. Second, even when the same exit nodes are selected, some exit nodes are incapable of handling 1,000 HTTP requests because of resource limitations.

Ethical Standards. The application layer scans comprise HTTP `GET` requests to Alexa top 1K websites directly, and through each of 900 exit nodes. Thus each target website receives 901 HTTP requests which are spaced out over time to avoid overwhelming the target. Note that the targets are globally popular websites, presumably capable of handling thousands of simultaneous requests. Each of the 900 exit nodes handled 1,000 HTTP requests. This probably overloaded some of the resource-limited nodes, potentially leading to degraded performance for other Tor connections sharing these nodes. We believe that the benefits provided by this study in exposing the prevalence and nature of Tor blocking outweigh the inconvenience experienced by some Tor nodes and users during the 5 days when this experiment was conducted. The probe comprises HTTP `GET` request which does not pose any security risk to the target. The probed list poses no risk to the authors running the experiment, comprising globally popular websites according to Alexa with no apparent bias towards controversial content.

We infer that a website discriminates against Tor when it responds with a `200` 'OK' status code when visited without Tor, and some other valid but non-`200` level status code when visited with Tor. On average, about 3.67% of the Alexa top 1K websites respond with a non-`200` status code when visited through Tor (Table 4.6), whereas only at most 6 sites respond with a non-`200` code when visited without Tor. To check whether this difference is statistically significant, we compute the p-value using permutation test under the null hypothesis of independence. We choose the permutation test because it does not assume that the responses of the experimental units are independent and identically distributed: many of the top Alexa top 1K websites that we test are hosted on CloudFlare and Akamai CDNs, whether all the tested websites send a non-`200` response to a Tor exit is not independent. The p-value of the permutation test is 0.008 which shows that Web visits through Tor receive different treatment from the websites. We also encountered around 8% non-HTTP errors such as timeouts and connection resets, which can be caused

by discrimination at layers 3 and 4.

We find that no site blocks all Tor exit nodes (Figure 4.4). During the five days when we conducted our experiment, on average 15.6 sites out of the Alexa top 1K top sites blocked over 60% of Tor exit nodes. These sites include `yelp.com` (up to about 70% exit nodes blocked), `macys.com` (up to $\approx$ 60%), and `bestbuy.com` (up to $\approx$ 66%). The majority of these sites are hosted on Akamai and Amazon Web Services. All the websites on Akamai show a `403` 'Access Denied' block page, which cannot be bypassed. Yelp and Craigslist have their own block page. Some websites such as `macys.com` return a redirect error that often leads to an infinite redirect loop. On average, around 69 sites block over 10–50% of the Tor exit nodes. The majority of these websites are hosted on CloudFlare. On average, around 150 websites block less than 10% of the Tor exit nodes and the rest of the websites (over 700) do not block any exit nodes at the home page.

To check whether these blocking events are abuse-based or Tor-specific, we look at the age and exit probability of exit nodes. We assume that abuse-based blocking is more likely to block old or high-probability exit nodes because these have more opportunity to attract abuse, while blacklist-based or Tor-specific blocking tends to block all exits equally. We download node exit probabilities and ages from Onionoo [165]. We use logistic regression to determine the effect of the exit characteristics on blocking rate.[2] Overall, we did not find any statistically significant effect across all the measured sites; however, for specific websites and specific blockers we find significant effects. We manually test three sites: `bestbuy.com` (on Akamai), `change.org` (on CloudFlare), and `4chan.org` (on CloudFlare with an apparently lower security configuration). For `bestbuy.com`, both exit probability (odds ratio = 2.4 per 1% change in exit probability with p-value = 0.0098) and age (odds ratio = 1.002 per day of age with p-value < 0.001) have an effect on blocking frequency. For `change.org`, the effect of exit probability is not significant, but age has a slightly greater effect (odds ratio = 1.003 per day with p-value < 0.001) than with the Akamai-hosted `bestbuy.com`. For `4chan.org`, exit probability has a moderate effect (odds ratio = 1.9 per 1% with p-value < 0.001) and age has an even greater effect (odds ratio = 1.004 per day with p-value < 0.001) than `change.org` hosted on CloudFlare. We observe that the two CloudFlare-hosted sites do not block any exits younger than about 30 days, while the Akamai-hosted site does. Figure 4.5 compares the different subsets of exits blocked by the three sites.

Some sites do not block Tor users from accessing their home page, but prohibit accessing 'deeper' pages or functions. We conduct a small ancillary experiment on search engines with URLs containing search queries (as opposed to the URLs of home pages). The home page of `google.com` is not blocked from any Tor exit node, but searching on Google is blocked from 23–40% of the exit nodes (varying between different days). We notice similar

[2] We acknowledge that logistic regression requires each observation to be independent which might not be true in our case. We chose logistic regression because it can handle non-linear relationships and can provide an estimation of the effect.

behaviour for Yahoo!, where searching is blocked for around 1% of the exits, but the homepage is always accessible.

We find 42 exit nodes that are not blocked by any of the Alexa top 1K websites during our crawl spanning 5 days. These exits do not appear to be dedicated Tor exit nodes. All the exits have similar characteristics: *(i)* all except one are hosted on Amazon EC2, *(ii)* their node bandwidth is unmeasured, *(iii)* all the exits are turned on and off periodically, and were never up consecutively for more than a month, *(iv)* most of exits are started and stopped at the same time. We suspect that these exit nodes are unblocked either because of their low bandwidth (20 KB/s) or low uptime.

4.4.2 Historical Perspective from OONI

For a historical record of Tor blocking, we draw upon scan data published by OONI, the Open Observatory of Network Interference [155]. Volunteers run a program called ooniprobe that runs a variety of network tests and reports the results to a central collector. The network tests are designed to detect behaviour such as DNS tampering, blocking of anticensorship proxies, and manipulation of HTTP headers. The oldest reports are from December 2012, and these continue to the present.

One of ooniprobe's several tests, http_requests [166], suits our purpose of detecting differential treatment of Tor users. The test takes as input a list of URLs and downloads each URL twice, once with Tor, and once without (both downloads happen within a few seconds of each other). The results consist of a set of Tor–non-Tor *request pairs*. Each request in a pair maps to a response, either an HTTP response with status code, header, and body; or an indication that an error occurred such as a timeout or rejected connection.

The http_requests test was intended to discover blocking by the local network, with the Tor request serving as a control (uncensored request). We turn the intended methodology on its head, using the non-Tor request as a control and observing how the response to the Tor request differs. Within a single execution of http_requests, each URL is downloaded through Tor once, through a single exit node. The same exit node is reused for multiple URLs, but changes over time (even within a report) as circuits are naturally rotated. Path selection favors exit nodes with higher bandwidth (Figure 4.3), meaning that larger exit nodes get tested more often. However, the large number of available OONI reports means that all but the rarest exit nodes receive at least some representation.

The list of tested URLs varies across reports. For the most part, ooniprobe uses the Citizen Lab URL testing lists [167], which consist of about 1,200 'global' URLs, in addition to up to about 900 additional country-specific URLs that depend on the geographical location where ooniprobe runs. (Versions of ooniprobe before October 2014, test a static list of 1,000 URLs derived from the Alexa top sites.) There are also reports that use a manually specified URL list. Therefore some URLs are tested more often than others. We only consider URLs that are tested at least 100 times.

Ethical Standards. Our historic analysis of Tor blocking draws on scan data published by OONI, the Open Observatory of Network Interference [155], a free platform under the Tor project. Volunteers run OONI to uncover censored content by probing a list of websites, flagging ones that cannot be directly fetched but are accessible over Tor. The probe list comprises controversial sites including pornography and hate speech, instant messaging applications, and circumvention tools. This poses a number of risks to the volunteers—they have been advised to run the tool at their own risk according to OONI's software license [168]. Depending on the local law, volunteers can be held liable for the use of network measurement software, censorship detection tools, censorship circumvention software, or for accessing certain websites. These activities can be potentially detected by third-parties (e.g. the government, the ISP, or employers) through network surveillance, or direct access to volunteers' machines. Moreover, the results of OONI probes are sent to a measurement collector and automatically published (unless users opt out), potentially revealing IP addresses or other identifying information.

Up through 20 July, 2015, the raw OONI http_requests data consist of 2,505 reports, 2,574,326 Tor–non-Tor request pairs, and 102,865 distinct URLs. We apply a number of restrictions to the raw data to obtain a subset useful for our analysis:

- We discard reports before September 2014. Reports after this date (82% of the total) occur more regularly than before.

- We discard URLs with a small number (<100) of request pairs. The great majority of distinct URLs are tested only a handful of times and thus not appropriate for our analysis. Although only 2% of URLs occur often enough, these account for 89% of all request pairs.

- We discard request pairs where one or both responses are missing. A response to an http_requests probe is either an HTTP response (i.e. with a status code such as `200`) or an indication of timeout or rejection. About 20% of request pairs are anomalous, missing a response data structure; but these are concentrated in a tiny fraction of reports and URLs.

In brief, we seek URLs that have been frequently sampled, at close time intervals, that have meaningful response data. After applying all these restrictions together, there remained 1,969 reports, 1,727,138 request pairs, and 2,387 unique URLs.

Our basic analysis technique compares the Tor and non-Tor responses in each request pair. We specifically look for cases where the Tor request is blocked and the non-Tor request is unblocked. We consider a URL 'blocked' if the request: *(i)* timed out, *(ii)* was rejected, or *(iii)* received an HTTP response with status code `400` or higher. We treat redirect status codes like `302` as 'unblocked'. A limitation of this approach is that we might miss the cases where Tor is blocked by redirection to a block page. We also treat certain other responses as special cases such as HTTP `408` 'Request Timeout', which occurs when

the client does not send its request in time, and is more likely a measurement error than blocking. This methodology of classifying responses by status codes is crude—but it is tenable precisely because we have paired simultaneous Tor and non-Tor responses. If a Tor request receives an HTTP `403` 'Forbidden' response, it does not in itself indicate differential treatment of Tor users. But if, at the same time, a non-Tor request receives an HTTP `200` 'OK' response, it serves as evidence that the server treats Tor users differently. If both requests time out, say, or both succeed, then we do not consider it an instance of discrimination against Tor. For our purposes, we consider the case where Tor is unblocked and non-Tor is blocked (which is what the OONI http_requests test is meant to find) as both being unblocked (i.e. no negative Tor discrimination). This method of comparing paired responses does away with some of the difficulties in distinguishing variations that arise due to blocking and benign variations based on geolocation, for example.

There are some limitations to our approach. Sometimes servers return block pages with a non-error status such as `200`; we miss such cases. The results possibly partially conflate Tor blocking with general anti-bot blocking; that is, some blocks may be because of Tor, and others may be because of ooniprobe. We suspect this is the case for `www.amazon.com`, for example (discussed later). Some installations of ooniprobe run in censored places. Because of how we count responses, in the worst case we miss an instance of Tor discrimination (because Tor and non-Tor both appear to be blocked).

We now quantify the amount of blocking we observe in the OONI data. First, we give the overall rates of Tor versus non-Tor blocking. Recall that each request pair consists of a Tor and a non-Tor request, each of which may be blocked or unblocked, leaving four possibilities. The highlighted row is our focus of interest:

84.4%	Both requests unblocked
6.8%	Tor request blocked only
1.8%	Non-Tor request blocked only
7.1%	Both requests blocked

Drilling deeper, we find that a little more than half of the 6.8% Tor blocking happens at the application layer; that is, block pages served as HTTP responses. The other blocks are transport layer rejected connections and timeouts.

6.8% =	0.45%	rejects	} transport layer
	+ 2.82%	timeouts	
	+ 3.54%	HTTP	} application layer

Finally, we list the organizations that are responsible for the most Tor blocking. To categorize blockers, we write regular-expression classifiers and run them against the OONI HTTP responses. Together these constitute the 3.54% figure in the previous table.

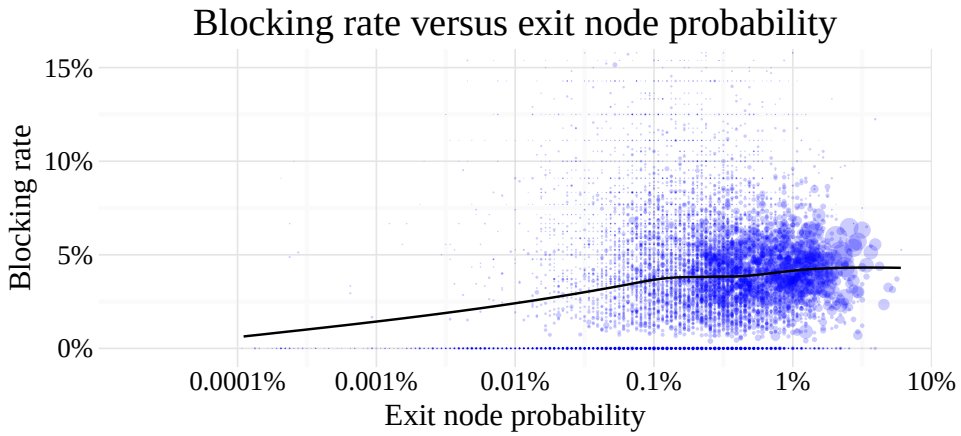


Figure 4.6: As Tor exit probability increases, so does the incidence of blocking. Each dot represents a single exit node and its rate of being blocked, as seen from nearly a year of OONI measurements. The dark line shows a smoothed mean of the blocking rate. For clarity, the graph omits some points with a blocking rate above 15%; these constitute only about 0.5% of the data mass.

2.507%	CloudFlare (CDN)
0.362%	other
0.349%	custom
0.144%	Bluehost (web host)
0.126%	Akamai (CDN)
0.028%	Site5 (web host)
0.028%	Convio (web host)

CloudFlare is a content delivery network that offers an abuse-based blocking system (turned on by default) that, when tripped, forces the user to complete a CAPTCHA before continuing to the site. The next row, marked 'other', includes all pages for which we do not write a specific classifier. The 'custom' row encompasses a wide variety of bespoke block pages belonging to one specific web site: sites in this category include Craigslist and Yelp. Bluehost is a web hosting company. Akamai is a content delivery network. Site5 and Convio are web hosting companies.

Figure 4.6 shows that blocking rate increases proportionally with exit probability. Figure 4.7 illustrates the potential impact of a large centralized provider. Here, the blocking rate of CloudFlare sites suddenly drops, while other forms of blocking remain unchanged. This means it is possible for one company to have a unilateral effect on many users' browsing experience.

A small number of block pages explicitly target Tor users. The hosting company Convio sends a simple `501` (Not Implemented) page that says, "Not Implemented Tor IP not allowed" and offers no opportunity to continue. The site `ezinearticles.com` serves a custom `403` (Forbidden) block page that says, "it appears that you are using

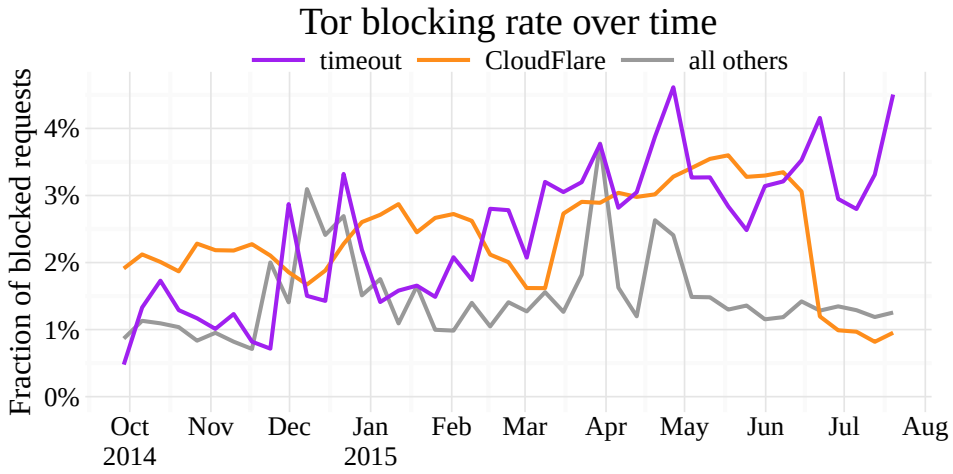
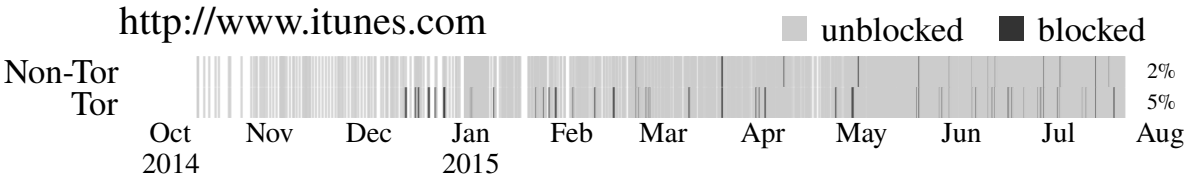


Figure 4.7: Tor blocking rates over time. We have separated out CloudFlare blocks to illustrate both the fact that CloudFlare is the most common blocker (at least among the URLs in the OONI set), and how CloudFlare's rate of blocking decreases, possibly reflecting a policy change.

Tor anonymizing software. No Problem! We just need you to enter a Captcha so we can confirm that you are a person and not a bot". The site permits browsing after solving the CAPTCHA.

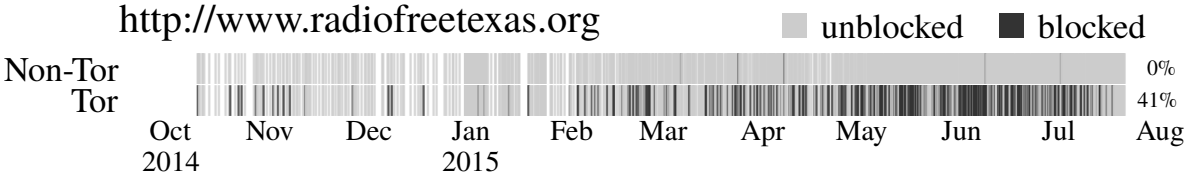
We conclude this section with a sampling of time series that compares the patterns of Tor and non-Tor blocking for selected URLs. These URLs exemplify various types of blocking. It is possible to distinguish sites that employ a Tor blacklist, because these have near 100% rates of Tor blocking. We can readily link sites that share a CDN or web service provider by temporal patterns in their blocking. In the charts below, each request pair corresponds to a vertical strip across two rows, one for the non-Tor request and one for the Tor request. A light shade in the row means the request is unblocked and a dark shade means it is blocked.

Blocking as a whole is not all that common. Most URLs manifest like this one, where potential blocking is scattered, intermittent, and rare:

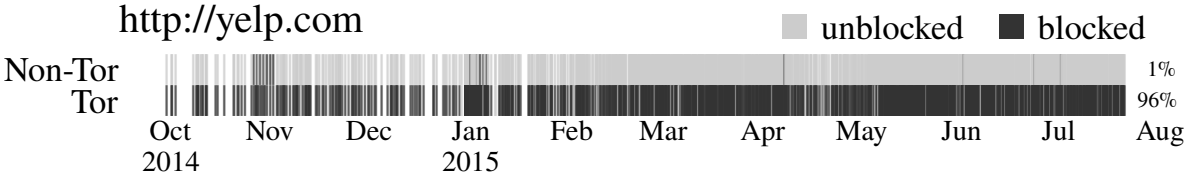


In contrast, here is an example of a site running on CloudFlare, the largest source of blocking. Non-Tor is almost always accessible, while Tor is often—although not always—blocked. We have found that for sites such as this one, simply retrying the request with a different Tor circuit often makes the site accessible.

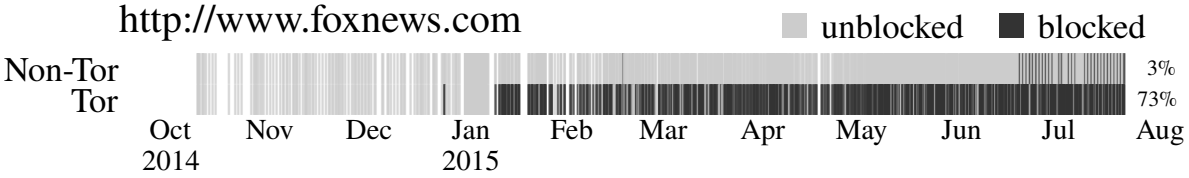
Another common blocker is a web hosting service called Site5. It also disproportionately blocks Tor visitors, though not at as high a rate as CloudFlare does.



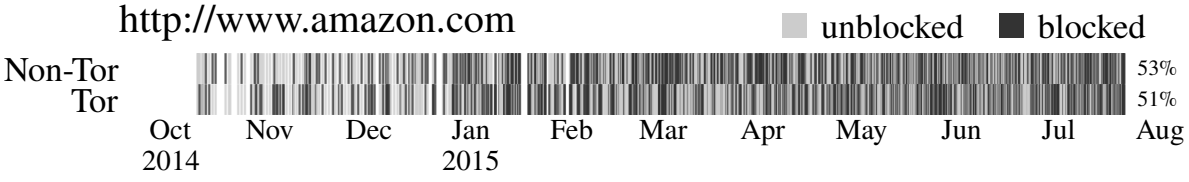
There are a few sites that evidently employ a blacklist of Tor exit nodes. Their rate of Tor blocking is nearly 100%.



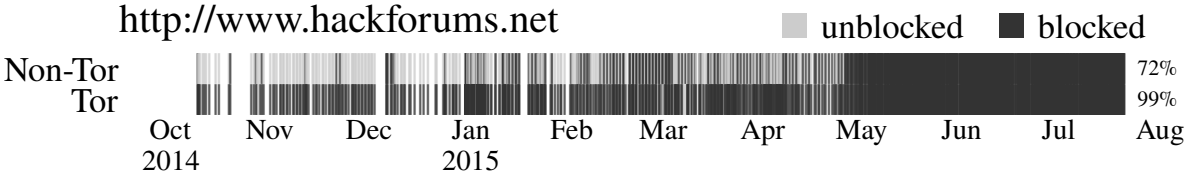
Similarly, some sites now have Tor blocks, but previously allowed Tor. The server `www.foxnews.com` serves an Akamai block page to all Tor clients, but only began to do so in January 2015.



The web server at `www.amazon.com` is an interesting case because of its nearly equal blocking rates of both Tor and non-Tor traffic. We suspect that this kind of blocking is not in terms of abuse or proxy blocking in itself, but rather probabilistic anti-bot or anti-crawling detection; in this case, it detects ooniprobe as not corresponding to a human with a browser. The text on the block page supports this idea: "To discuss automated access to Amazon data please contact..."



The site HackForums.net used to block both Tor and non-Tor visitors, Tor at a higher rate. In May, 2015, the site announced on Twitter that "Most countries aggressively blocked again. Sorry but the attack traffic, scammers and spam are not worth it". All of ooniprobe's requests became almost completely blocked. This reflects the website deploying anti-bot or anti-crawling detection to mitigate crawling and spamming by bots.



4.5 Discussion

Based on our measurements, we find instances of both blocking all traffic from Tor exits and cases of fate-sharing, where ASNs and websites block Tor exit traffic due to employing automated abuse-based filters. In the case of entities that preemptively block all Tor exit traffic, there is little that can be done beyond detecting instances of this occurring and publicizing the entities that do so. With abuse-based blocking, the potential for more precise filtering could enable benign users to avoid blocking that targets the abusive actions of other users. In this section, we discuss several potential methods of reducing this filtering, more precise abuse-based filtering, and minimizing the impact of this growing threat to people that use anonymity networks such as Tor.

4.5.1 Anonymous Blacklisting Systems

Anonymity networks such as Tor cloak the user's true IP address, making it difficult for ASNs and website operators to differentiate abusive users from benign visitors based on IP address. This causes many automated and list-driven abuse-detection systems to blacklist some or all of the exit nodes' IP addresses.

Anonymous blacklisting systems have been proposed as a method to enable website operators to more precisely allow benign visitors to access their sites and hold abusive users accountable for their actions [169] [170]. The goal of anonymous blacklisting systems is to allow a website, such as Wikipedia, to block access to an individual anonymous abusive user without requiring a trusted third-party that can revoke a user's anonymity. This capability would allow websites to defend themselves against anonymous abusive users using similar methods as against identifiable users. Most anonymous blacklisting systems require users to anonymously register and authenticate with the blacklisting service using blind signatures or zero-knowledge proof techniques, and create whitelists of permitted users. The registration process must require anonymous payment or otherwise bind users to scarce resources, such as IP addresses, to mitigate Sybil attacks.

Adoption of anonymous blacklisting systems has been negligible due to issues of degraded user privacy—anonymous blacklisting systems either offer pseudonymity instead of full anonymity or require a semi-trusted third-party to provide anonymity—and additional computational overhead [171]. If these issues could be addressed, anonymous blacklisting systems might be more widely deployed by anonymity networks and website operators, reducing the amount of explicit anonymity network blocking and fate-sharing experienced by users of these systems.

4.5.2 Contextual Awareness

It is conceivable that anonymity networks could reduce instances of abuse-based filtering by learning which websites were blocking certain exit nodes and reroute requests for these

sites to another exit node that is not blocked. This would likely require application layer analysis on the exit node that might be overly invasive from a privacy context.

A less privacy-invasive technique could involve the Tor Browser displaying a message when filtering is suspected. This could be done by including block page detection similar to that used in our study. The browser could also offer to retry the request using a different exit node. Both of these techniques could marginally reduce the impact of abuse-based blocking. However, neither of these approaches directly addresses fate-sharing issues caused by abuse-based blocking of Tor exit IP addresses. These approaches also have the potential to trigger an "arms-race" as abusers could benefit from Tor spreading out abusive traffic causing more aggressive filtering of Tor traffic by impacted services.

4.5.3 Redesigning Anonymity Networks

Tor and other anonymity networks could attempt to recruit a larger pool of exit nodes that enables each exit to deliver a smaller amount of traffic. Our results find a (weak) correlation between the amount of traffic a node exits and the probability of a node's IP address being blocked due to automated abuse-based filtering. Thus, reducing the amount of traffic each node exits might reduce their probability of being blocked by automated abuse-based filtering. The risk of this and other techniques to fan out traffic to more IP addresses is that it might cause more websites to preemptively block *all* Tor exit traffic. This also does not deter abusive use of Tor.

We could also consider disincentivizing large-scale abuse by charging Tor users for traffic usage. The BRAIDS system proposed an anonymous payment scheme for improved quality-of-service originally with the goal of disincentivizing users from performing bulk downloads using Tor [172]. BRAIDS could also be used to charge Tor users for traffic usage. This might reduce the amount of abuse, but at the cost of Tor becoming unusable by people that are not willing or cannot pay for usage or improved quality-of-service.

4.5.4 Redesigning Automated Abuse Blocking

Basing automated abuse blocking on ratios of abusive to benign requests instead of absolute values might reduce the instances of higher-bandwidth exit nodes being blocked by abuse-based filtering. However, this would allow abusive users to insert benign chaff requests to evade automated abuse filtering based on ratios instead of fixed limits.

Another idea is to never completely block requests and instead display CAPTCHAs to low-reputation IP addresses associated with Tor exit nodes. The risk of websites not blocking Tor exit node IP addresses is that CAPTCHAs are an economic deterrent to large-scale abuse that might be insufficient in cases of profit-driven abuse, such as spam [173]. This highlights the challenges of websites that block Tor exit node IP addresses in self-defense based on automated abuse filtering systems.

4.6 Summary

In this chapter we studied publisher-mandated blocking in the context of the Tor anonymity network, where a large number of websites provide Tor users with degraded service; resulting in them effectively being relegated to the role of second-class citizens on the Internet. We contributed a systematic methodology and measurement study of the scale of blocking of anonymity networks, both at the network layer and the application layer. We measured that at least 1.3 million addresses in the IPv4 address space, and approximately 3.67% of the Alexa top 1K websites, either block or offer degraded service to Tor users. We provided a first step in illuminating the scale of the problem and identified centralized mechanisms that impact the usability of many sites for users of anonymity networks. We identified two kinds of network layer blocking: wholesale blocking by Autonomous Systems (ASes) such as access ISPs, and more targeted (likely abuse-driven, and thus implicit) blocking practiced by content hosting sites and service providers. We note that while many websites block Tor to reduce abuse, doing so inadvertently affects users from censored countries who do not have other ways to access censored Internet content. After being presented at The Network and Distributed System Security Symposium (NDSS) in February 2016, this study gained media coverage [174], [175]. This led to a dialogue between Tor developers and CloudFlare, one of the top Tor blockers identified by our results, to find solutions to offer Tor users a decent browsing experience [176]. In March 2016, a month after this work was presented, CloudFlare allowed its customer website owners to specify rules that are applied to Tor traffic in the same way as traffic from a country: they can whitelist Tor traffic or get Tor users to solve CAPTCHAs, but they cannot fully blacklist Tor [177].

Generality of Research Methodology and Findings. The methodology presented in Section 4.3 to assess network layer discrimination of Tor can be applied to measure discrimination of other applications if the feature that triggers publisher-side blocking is known (e.g. in Tor's case, this feature is IP addresses of exit nodes). Ideally, this feature should be unique to the application being studied to establish with confidence that publisher-side blocking specifically targets the application being studied. We note that measurement is more feasible if publishers block the application based on user-side features such as the presence of specific software instead of IP addresses which requires physical access to such machines. The methodology provides confidence in Tor blocking through consistent lack of response to probes from Tor exit nodes vs. control nodes, implying that the experiment has to be repeated over multiple consecutive days. Another limitation is the choice of our control nodes which are all based in universities. IP addresses associated with universities have a clean reputation and are often whitelisted, so what we perceive as Tor blocking could represent policy to block low reputation IP addresses.

Section 4.4.1 presents a methodology to measure application layer discrimination of Tor by sending HTTP `GET` requests to Alexa top 1K URLs once from all available Tor exit nodes (using Exitmap [161]) and once without Tor (using a custom Python script). Tor

exit nodes may come up and down while the experiment is in progress. This effect can be minimized by running the experiment over a shorter time interval by running multiple instances of Exitmap in parallel. Our crawl, comprising 5 instances of Exitmap to send HTTP requests to Alexa top 1K URLs through each of 900 exit nodes, takes 10 hours to finish. Because of churn in the availability of Tor exit nodes, and the inability of some exit nodes to handle 1,000 requests, we saw an average of 800 requests per exit node. Future studies should adjust these parameters according to the number of URLs being scanned, the number of exit nodes through which the probes are being sent, and the extent of scan parallelization while taking care not to overload Tor network. We note that this methodology only reveals Tor blocking at the granularity of index pages.

We measure application-layer blocking of Tor across Alexa top 1K URLs. Effectively, the scope of our results is restricted to this list. Alexa maintains a list of most popular websites, globally and by countries, using an opaque methodology that combines a site's estimated average of daily unique visitors and its estimated number of pageviews over the past 3 months relative to all other websites over the Web (or in a country for regional popularity). These estimates are based on data collected from a large number of Internet users that use one of Alexa's over 25,000 different browser extensions, and websites that run the Alexa script. It is suspected that Alexa's ranking methodology outweighs the WEIRD (Western, Educated, Industrialized, Rich, and Democratic) population [178]. If this is the case, then our results represent prevalence of Tor blocking across websites that cater to the so-called WEIRD population.

Section 4.4.2 presents historic blocking of Tor by drawing on scan data published by OONI. As the original goal of this data is to measure censorship, the list of probed URLs is biased towards sensitive, potentially censored topics, which is not the best sample to study global prevalence of Tor's publisher-side blocking.

Both our active and passive methodologies to measure application-layer blocking of Tor cannot detect instances where a website blocks Tor by displaying a block page with `200` status code or by redirection to a block page with the code `302`. Finally, all the methodologies presented in this work provide a lower bound on instances of blocking: firewalls can silently drop probes at the network layer, and at the application layer websites can ignore requests from automated crawlers.

Data Release. All relevant data, code, and auxiliary information are available from the University College London database, under the DOI: `http://dx.doi.org/10.5522/00/5`.

Chapter 5

Differential Treatment of Adblock Users

In this chapter we analyze publisher-side blocking in the context of users of adblocking software. Today's Web ecosystem is largely driven by online advertising. However, recent years have seen a large number of users turn to adblocking and tracker-blocking tools[1] for the purposes of improving their Web browsing experience, maintaining privacy, and more recently to protect themselves against malware [179] [180]. With a recent study estimating the number of active adblock users to be 198 million and revenue losses due to adblockers at $22 billion [181], the threat posed by adblockers to the online advertising revenue model has moved from mildly concerning to existential. In response, publishers have started to actively detect users of adblockers, and subsequently block them or otherwise coerce them to disable the adblocker—in the rest of this chapter, we refer to these practices as *anti-adblocking*. Most recently, this practice gained wide attention with the endorsement of the Internet Advertising Bureau (IAB) when, in March 2016, it released a primer on how to deal with users of adblockers, as well as a semi–open-source script made available to members of the IAB, for detecting the use of adblockers [182]. The tension between key stakeholders in this ecosystem—publishers, users, and a plethora of intermediate beneficiaries—forms part of what has been dubbed as the *adblocking arms race* [183].

In this chapter, we characterize anti-adblocking practices across Alexa top 5K websites. We discuss related work in Section 5.1. In Section 5.2, we develop a scalable technique to identify popular third-party services that are shared across multiple websites. We employ this methodology to flag anti-adblocking scripts and understand how these operate, mapping out the entities that serve anti-adblocking scripts and the websites that use these scripts (Section 5.3). We conclude with a discussion of the anti-adblocking arms race in terms of ethics and legality, also enumerating existing proposals that aim to achieve a sustainable and unintrusive online advertising model (Section 5.4). Section 5.5 concludes the chapter and provides information about how to access source code and data.

[1] While adblocking differs from tracker-blocking, to ease presentation, we refer to tools that provide any of these properties as adblockers.

5.1 Related Work

Rafique *et al.* [184] measure anti-adblocking as an incidental aspect of a broader study of malicious and deceptive advertisements, malware, and scams on *free live-streaming services*. They find that anti-adblocking scripts were used by 16.3% of the 1,000 domains they crawled, which is a bit higher than what we find in the Alexa top 5K (6.7%), although not surprising given their heavy use of deceptive ads.

Our study also complements work quantifying and characterizing non-transparent third-party Web services, as well as revealing users' differential treatment. For example, Ikram *et al.* [185] proposed a machine learning approach to characterize JavaScripts used for online tracking and those used for providing website functionality. Their work allows privacy-enhancing tools to more selectively block JavaScripts without breaking website functionality. Acar *et al.* [186] and Liu *et al.* [187] measure the prevalence of tracking across large datasets of websites, while Mayer [188] studies the effectiveness of some adblocking and anti-tracking tools against those sites. Khattak *et al.* [189] assess discrimination against Tor users at the network and application layer. Various studies investigate price discrimination [190] [191] and its methods [192] employed by online marketplaces, and there are other studies on *filter bubbles*—the effect where high Web personalization leads to users being locked in information silos [193] [194].

All of these studies illuminate the nature and scale of opaque practices on the Web, informing our understanding of complex and multidimensional ecosystems. Our work complements previous studies by presenting a novel technique to identify shared objects across multiple websites at scale, and utilizing this approach to provide a first look at how the Web employs anti-adblocking techniques.

5.2 Methodology

This section presents our method for identifying third-party services that are shared between multiple websites. We describe the technique in the context of identifying shared anti-adblocking JavaScripts (JS). The premise of our approach is that by discovering *similar* objects (in our case, JavaScripts) that are loaded by multiple websites, we can infer the presence of a common third-party JS, its functionality, and its source.

Crawler Overview. We rely on a Selenium-based Web crawler to generate the set of JavaScripts to analyze. We load each website in our dataset with four browser modes: vanilla Firefox (with no extensions), Firefox with AbBlock Plus, Firefox with Ghostery, and Firefox with Privacy Badger. For each page load, we capture screenshots, HTML source code, and responses to all requests generated by the browser. We extract all the text between `<script>` and `</script>` tags from the HTML and label them as *embedded* JS. Similarly, we detect all JS objects in the collected responses and label them as *downloaded*

JS. In total, the top 5K Alexa websites generate over 200,000 individual JS files when loaded with the vanilla Firefox browser.

Ethical Standards. We download Alexa top 5K websites with four browser modes, thus each website receives four requests from us which can be easily handled without impact on performance. The request—a simple HTTP `GET`—does not pose any security risk to the target. The probed list poses no risk to the co-authors running the experiment, comprising globally popular websites according to Alexa with no apparent bias towards controversial content.

Identifying JS Objects with Common Sources. We formulate our problem of finding groups of similar JS as a maximal clique finding problem [195]. We consider each JS file loaded by a website to be a node in a graph. If two nodes are within some margin of similarity of each other (we define our similarity metric later), we say there is an edge between them. We extract classes of JS that have a common source by identifying all maximal cliques in this graph. By intentionally focusing on finding similar JS (rather than identical JS), we allow for the grouping of objects that differ only slightly because of website-specific identifiers, features, and properties.

Choice of Similarity Metric and Threshold. To add an edge between two nodes in the graph (i.e. to indicate that two JS files in two different websites are similar), we need to define a metric for similarity and a suitable threshold for this metric. To measure the *similarity* of two JS files, we use Term Frequency–Inverse Document Frequency (TF-IDF) to generate a vector of *keyword weights* for each JS file after filtering out JS reserved words such as `function` and `var`. We then use the cosine similarity metric to measure the similarity of the two keyword weight vectors. Similar approaches using both TF-IDF and cosine similarity have been used by the information retrieval community for topic identification and similarity checking of source-code [196] [197]. We note that this method is particularly well suited to our task compared to other string matching approaches for the following reasons.

- **Whitespace Insensitive.** Many websites perform script minification using different libraries, yielding different indentation and whitespacing practices. Our approach is unaffected by these complications.
- **Position Insensitive.** In scripts that have several functionalities (e.g. tracking and ad-block detection), the position of each specific function is irrelevant to the similarity score.
- **Reasonably Resistant to Noise.** Small changes (e.g. website specific identifiers) have little impact on the final similarity score.

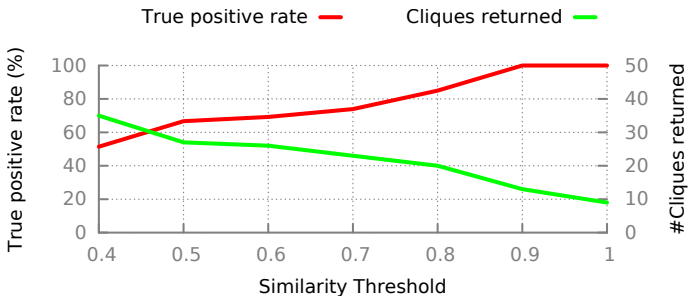


Figure 5.1: The effect of the similarity threshold parameter on the True Positive Rate (TPR) and the number of maximal cliques.

In order to determine a similarity score *threshold*, we perform a series of experiments on a small dataset of 4.4 thousand JS files extracted from the Alexa top 100 websites. In each experiment, we set a similarity threshold between 0.40 and 1.00 and compute the cliques in each of the corresponding graphs. We then manually inspect the cliques extracted at each threshold to identify the fraction of cliques containing JS with identical functionality and sources. Using this approach, we find that at a similarity threshold of 0.80, 17/20 cliques returned by our program contain scripts with identical functionality and sources; that is, achieving True Positive Rate (TPR) of 0.85. In Figure 5.1, we illustrate the change in TPR along with the number of cliques returned as the threshold increases. Although thresholds above 0.90 yield TPR=1.0, the number of cliques returned drops significantly, which will result in lower True Negative Rates (TNR). Therefore, following a conservative stance, we use a threshold of 0.80 for the remainder of our experiments.

Improving Scalability. Our approach involves computing the cosine similarity between each pair of keyword weight vectors, requiring $O(n^2)$ vector multiplications for n JS files. Given the large number of JS files used by websites (e.g. the Alexa top 5K sites contained over 200 thousand JavaScripts), this may not scale with large datasets. Therefore, we use the following heuristically developed filters to eliminate comparisons between scripts that are unlikely to be part of the same clique:

- **Word Count Filter.** We avoid comparing scripts with significant word count difference. Specifically, if a pair of scripts has a word count ratio higher than 1.50, we assume that these are unlikely to be a part of the same clique and set their similarity to 0.

- **Embedded vs. Downloaded Script Filter.** JavaScript is either embedded in the source HTML for page-specific functionality, or downloaded separately from external sources to provide site-wide functionality. We do not consider these as the same type of identity and set their similarity to 0.

- **Source Filter.** If two JavaScripts are fetched from the exact same URL, we mark them as identical.

	Cliques	Websites
Downloaded	1,373	3,619
Embedded	509	2,070
Trackers	456	2,741
Anti-Adblockers	22	335

Table 5.1: The number of total cliques (out of 1,882 found) and those related to tracking and anti-adblocking, along with the number of websites that incorporate these scripts (totalling 4,017 websites, computed over 200K downloaded and embedded scripts).

- **JavaScript Domain Filter.** JavaScript can communicate with external sources indicated by embedded URLs. We assume that for any pair of scripts, if one communicates with external sources and the other does not, their functionality is different and set their similarity score to 0.

Source and Functionality Identification. Once maximal cliques of similar scripts are identified, the content and meta-data of each script in a clique is used to generate and log: *(i)* the FQDN (Fully Qualified Domain Name) of the script's source, *(ii)* FQDNs of external resources utilized by the script, and *(iii)* keywords associated with the script. In Section 5.3, we use these three features, in addition to content of the script, to classify cliques by functionality.

Method Limitations. We acknowledge that our method has a few limitations. First, our similarity metric will fail to identify obfuscated JS code. Second, given that we do not compare downloaded with embedded JS code, we may fail to identify small cliques in which a reduced number of sites integrate an anti-adblocking JS in a different way than is normal. Finally, our method may fail to identify similarities between composed JS; that is, scripts that consist of multiple individual files downloaded as a single object. As a result, our method only provides a lower-bound approximation of the usage of anti-adblocking across websites. We plan on addressing these limitations in future work.

5.3 Dataset and Results

We apply our clique detection methodology to the JS objects fetched by our crawler using the vanilla Firefox browser. We restrict our analysis to cliques of size greater than 5 (i.e. JavaScripts shared by more than 5 sites in our dataset) as we are interested in identifying scripts that are shared across many websites. We acknowledge that this approach might fail to flag anti-adblocking scripts employed by individual or a small number of websites, and the scripts that are used by a few websites in the Alexa top 5K but popular among websites

ranked above 5K. As shown in Table 5.1, we find 1,373 cliques that are shared among 3,619 websites in the downloaded files, with an average of 232 websites per clique (σ=365.6) and the largest clique having 1,320 websites (which we find, via manual inspection, is a JS related to jQuery). Among the embedded scripts, 509 cliques are shared by 2,070 websites (μ=41.2 σ=48.9 max=261).

We manually analyze all the 1,882 cliques (corresponding to 4,017 unique websites) identified for both downloaded and embedded scripts. We tag the cliques as *trackers* (that upload information such as IP addresses and cookies to tracking companies), *anti-adblockers* (that check for the presence of adblockers), or *others*. We perform manual analysis by identifying external libraries and function specific keywords used in the scripts. Note that manual analysis of JS is a tedious process that does not scale to a large number of scripts; we will explore automated JS tagging in future work.

We uncover 22 cliques that are used for anti-adblocking. These cliques are employed by 335 websites—representing about 6.7% of the Alexa top 5K websites. We observe that the Alexa top 1K has 60 anti-adblocking websites, and the number increases by about 70 websites for every additional 1K considered, reaching 335 anti-adblocking websites in top 5K. While studying anti-adblockers, we also identify 456 tracking cliques employed by about 54% of the Alexa top 5K, validating previous studies on the pervasiveness of tracking over the Web [198].

Anti-adblocking by Website Categories. In Table 5.2, we report the categories of the 335 anti-adblocking websites using McAfee's URL categorization service [132]. We find that anti-adblocking is common among a diverse mix of publishers, and prevalent among publishers of "General News" (19.5%), "Blogs/Wiki" (9.3%), and "Entertainment" (8.5%) categories, which represent more than one third of all websites. Note that these categories are also among the most popular ones across all top 5K Alexa domains, although to a lesser extent—respectively, 9.4%, 6.29%, and 5.4%. Whereas, other popular categories among top 5K domains (e.g. "Internet services", "Online Shopping", "Business", which account for 20% of the top 5K) are much less prevalent in anti-adblocking websites.

Website Response to Detection of Adblockers. To assess how anti-adblocking websites behave once they identify adblockers, we look at all the screenshots taken by our crawler when using the vanilla Firefox browser with no extensions and the Firefox browser with AdBlock Plus enabled (which we assume is more likely to be detected due to its popularity [179]).

We note cases where there is an explicit (i.e. warning to disable adblocker) or a discreet (i.e. blank page via AdBlock Plus, but normal appearance without) response to adblocking. For these websites, we also view screenshots when accessed by the Firefox browser with Ghostery, Privacy Badger, and NoScript.

We find only 6 explicit and no discreet responses to adblocking. Of the explicit

%	Category	%	Category (continued from left column)
19.5%	General News	2.5%	Pornography
9.3%	Blogs/Wiki	2.5%	Forum/Bulletin Boards
8.5%	Entertainment	2.2%	Technical/Business Forums
4.3%	Internet Services	2.2%	Potential Illegal Software
3.7%	Sports	2.0%	Online Shopping
3.7%	Games	1.7%	Portal Sites
3.2%	Travel	1.7%	Humor/Comics
3.2%	Education/Reference	1.2%	Social Networking
2.7%	Business	1.2%	Provocative Attire
2.5%	Software/Hardware	1.2%	Marketing/Merchandising

Table 5.2: The distribution of anti-adblocking websites by category (according to McAfee's URL categorization).

responses, 3 are displayed by porn websites hosted by the same company, `MindGeek`, and employ the same anti-adblocking script downloaded from `DoublePimp`. The warning is displayed for both AdBlock Plus and Ghostery. The remaining 3 also employ the same script, but display different messages (only for AdBlock Plus) with the same general theme; that is, nudging the user to disable the adblocker or support the website via subscription or donation.

Some websites display adblocker warning to users after they engage in some form of activity such as clicking on links or scrolling. To capture such responses, we repeat the above exercise for screenshots taken after mimicking user activity; that is, clicking on a random link on the page, scrolling down to the bottom of the newly loaded page, waiting three seconds, then scrolling back up to the top of the page, and waiting 5 seconds. While the modified methodology validates our previous observations, we do not discover any new responses.

In the attempt of automating the analysis of websites' response to anti-adblocking, we also tried to use image comparison tools, such as perceptual hashing. However, this generates a high number of false positives due to dynamic content on many sites as well as false negatives since anti-adblocking warnings and messages generate a relatively small visual difference.

Anti-adblocking Mechanism. We manually inspect the 22 anti-adblocking scripts (14 downloaded and 8 embedded) aiming to understand how anti-adblocking scripts detect adblockers. We note that of these only the 14 downloaded scripts are actually useful as the 8 embedded scripts simply redirect to the downloaded scripts. We find that anti-adblockers operate on a simple premise: if a bait object (i.e. an object that is expected to be blocked

Domain	Description	#Websites	Adblockers		
			AdBlock Plus	Ghostery	Privacy Badger
pagefair.com	Anti-adblocking	20	✓	✗	✓
googleadservices.com	Ads	61	✗	✗	✗
googlesyndication.com	Ads	13	✗	✗	✗
taboola.com	Ads	36	✗	✓	✓
outbrain.com	Ads	10	✗	✓	✓
ensighten.com	Ads	6	✗	✓	✗
hotjar.com	Analytics	9	✗	✗	✗
doublepimp.com	Pornography	8	✗	✓	✗
tacdn.com	Travel	8	✗	✗	✗
cloudflare.com	CDN	50	✗	✗	✓
cloudfront.net	CDN	6	✗	✗	✗
ytimg.com	Content/Ads	108	✗	✗	✗

Table 5.3: The domains from which anti-adblocking scripts are downloaded and the number of websites employing them. The table's right side reports whether three popular adblockers counter-block anti-adblocking scripts from these domains.

by adblockers such as a JS or `DIV` element named `ads`) on the publisher's website is missing when the page loads, the script concludes that the user has an adblocker installed.

Specifically, the anti-adblocker detects adblockers by one of the following approaches: (1) The anti-adblocker injects a bait advertisement container element (e.g. `DIV`), and then compares the values of properties representing dimensions (`height` and `width`) and/or visual status (`display`) of the container element with the expected values when properly loaded. (2) The anti-adblocker loads a bait script that modifies the value of a variable, and then checks the value of this variable in the main anti-adblocking script to verify that the bait script was properly loaded. If the bait object is determined to be absent, the anti-adblocking script concludes that an adblocker is present. To track whether the user has turned off the adblocker after being prompted to do so, the anti-adblocker periodically runs the ad-block check and stores the last recorded status in the user's browser using a cookie or local storage.

Anti-adblocker Suppliers. We analyze the source code of the 14 anti-adblocking scripts and the domains from which these are downloaded aiming to infer the suppliers of these scripts. The remaining 8 embedded scripts redirect to anti-adblocking scripts served by `Cloudflare` and `Taboola`. Our analysis is summarized in Table 5.3. We also include a description of these domains (based on the information available on their official websites, Google search, and McAfee URL categorization service [132]) as well as the number of

websites in our dataset that employ the anti-adblocker.

At the top we find `Pagefair`, a company specialized in anti-adblocking services, followed by a number of domains related to `Google`, `Taboola`, `Outbrain` and `Ensighten`. Overall the anti-adblockers downloaded from these 5 domains are employed by 48% of all the 315 websites employing anti-adblockers. We note that these domains are direct beneficiaries of anti-adblocking as these inherently thrive on the prevalence of online advertisements. Though not directly related to online advertisement, the ability to detect adblockers is a useful capability for the analytics company `HotJar`.

We also find two cases where the anti-adblocking script is shared by entities in the same domain or business: `TripAdvisor` (*tacdn.com*) distributes the script to its 8 websites with different country code top-level domains. Adult websites, all of which are hosted by `MindGeek`, turn to `DoublePimp` for anti-adblocking. Two anti-adblocking scripts are pulled from popular Content Delivery Networks (CDNs), but we could not determine their original supplier. Finally, `ytimg` (a content server associated with YouTube) serves a script that has the ability to detect if ads were properly loaded, however, it is not clear how it uses this information.

Adblocker Response to Anti-adblocking. There is anecdotal evidence that the adblocking arms race has entered the next level: some adblockers can detect anti-adblockers and counter-block them [199]. To test for this behaviour, we visit a sample website for each anti-adblocking script via AdBlock Plus, Ghostery and Privacy Badger over the Chrome web browser. We repeat the experiment three times and monitor all HTTP requests generated when loading the website using Chrome's *Developer Tools*. We infer that the adblocker can counter-block if the request to fetch anti-adblocking script fails to be initiated. As reported in Table 5.3, half of the 12 anti-adblocking suppliers are blocked by at least one adblocker. Ghostery and Privacy Badger detect 4 anti-adblockers each, while AdBlock Plus detects only 1. Anti-adblocking scripts served by `Taboola` and `Outbrain` are blocked by both Ghostery and Privacy Badger, `PageFair` scripts by both AdBlock Plus and Privacy Badger, while `Doublepimp`, `Ensighten` and `Cloudflare` scripts by at most one of the three adblockers. We note that the anti-adblocking suppliers that are never detected are related to content distribution, Google ad services, analytics, or site-wide scripts.

5.4 Discussion

The adblocking arms race involves a plethora of players: Between publishers and consumers, a jostling array of intermediaries compete to deliver ads, mostly supported by business models that involve taking a cut of the resultant advertising revenue. At the heart of this rich ecosystem lie important questions regarding the legality and ethics of adblocking and anti-adblocking.

The legality of adblocking is potentially contestable under laws about anti-competitive business conduct and copyright infringement. To date, only Germany has tested these arguments in court, with adblockers winning most [200], but not all of the cases [201]. On the other hand, anti-adblocking in the EU might in turn breach Article 5(3) of the Privacy and Electronic Communications Directive 2002/58/EC, as it involves interrogating an end-user's terminal equipment without consent [202].

Many consider adblocking to be an ethical choice for consumers and publishers to consider from both an individual and societal perspective. In reality, however, both sides have resorted to radical measures to achieve their goals. The Web has empowered publishers and advertisers to track, profile, and target users in a way that is unprecedented in the physical realm [198]. In addition, publishers are inadvertantly and increasingly serving up malicious ads [180]. This has resulted in the rise of adblocking, which in turn has led publishers to employ anti-adblocking. The core issue is to get the balance right between ads and information: publishers turn to anti-adblocking to force consumers to reconsider the default blocking of ads for earnest ad-supported publishers, but defaults are difficult to shift at scale. Nevertheless, those publishers will fail if they do not redress in a fundamental way the reasons that brought consumers to adblockers in the first place. There exist proposals to provide a compromise, such as privacy-friendly advertising [203] as well as mechanisms to give users more control over ads and trackers that they are exposed to [188] [204]. Our work extends these efforts by providing quantified insights into anti-adblocking, to inform policy that can improve upon the current blocking–counter-blocking deadlock.

5.5 Summary

We studied publisher-side blocking of users of adblocking software. Adblocking tools continue to rise in popularity, potentially threatening the dynamics of online advertising. In response, a number of publishers have ramped up efforts to develop and deploy mechanisms for detecting or counter-blocking adblockers (which we refer to as *anti-adblockers*), effectively escalating the online advertising arms race. We developed a scalable approach for identifying third-party services shared across multiple websites, which was employed to map websites across the Alexa top 5K that perform anti-adblocking as well as the entities that provide anti-adblocking scripts. Our study revealed that at least 6.7% of Alexa top 5K websites conduct some form of anti-adblocking by downloading 14 scripts from 12 unique domains, most of which belong to ad services, while one specifically offers anti-adblocking services. Most of the anti-adblocking websites represent popular categories such as news, blogs, and entertainment. We studied the modus operandi of anti-adblocking scripts and manually visited sample websites from the anti-adblockers. We found that the arms race has already entered the next round—at least one of the three popular browser extensions (AdBlock Plus, Ghostery, Privacy Badger) could counter-block half of the

anti-adblocking scripts.

Generality of Research Methodology and Findings. The methodology presented in Section 5.2 to identify third-party services that are shared between multiple websites is generalizable—identifying trackers, UI scripts, and JavaScript libraries—in addition to anti-adblocking scripts which are the topic of this study (the limitations of this methodology have been listed at the end of Section 5.2). The downside of restricting analysis to shared services is that we fail to detect anti-adblocking scripts that are employed by individual or a small number of websites.

As we study anti-adblocking across Alexa top 5K, the scope of our findings is restricted to this list, which has well-known limitations as described in Section 4.6. Similarly, we our classification of anti-adblocking websites by categories inherits the accuracy of McAfee's URL classification service.

We manually label all the 1,882 shared services identified by our methodology to extract anti-adblocking scripts. This is a tedious process that does not scale to a large number of scripts; future work should explore automated methods to classify JS by functionality. To study the response of anti-adblocking websites after detecting adblockers, we look at all the screenshots taken by our crawler when accessing the websites with and without an adblocker. This process is also manual, but viable because of the small number of websites to analyze (335)—for large number of websites, an automated approach should be employed.

Source Code and Data Release. The source code of our JS clique extraction approach can be found at `https://bitbucket.org/rishabn/ad-study-code`. Data created during this research is available from the University of Cambridge data archive at `http://dx.doi.org/10.17863/CAM.703`.

Chapter 6

Conclusions

"The time has come," the Walrus said,
"To talk of many things:
Of shoes—and ships—and sealing-wax—
Of cabbages—and kings—
And why the sea is boiling hot—
And whether pigs have wings."

Lewis Carroll, "Through the Looking-Glass and What Alice Found There"

This dissertation analyzed the role of the censors' blocking choices in leaving behind a detectable pattern in network communications, that could be leveraged to detect and characterize censorship. Section 6.1 presents chapter-wise summary and key insights from this dissertation. Section 6.2 discusses potential avenues for future work and concludes the dissertation.

6.1 Summary and Insights

Chapter 2 sketched a comprehensive attack model to set out a censor's capabilities; with a discussion on the scope of censorship and the dynamics that influence the censor's decision. This was followed by an evaluation framework to systematize censorship resistance systems (CRSes) by their security, privacy, performance, and deployability properties; mapping these systems to the censor's attack model. Section 2.7 discussed the overarching research gaps and challenges in the area of censorship and its resistance. The remainder of the dissertation analyzed censorship in two contexts; user-side censorship and publisher-side censorship.

6.1.1 User-Side Censorship

Chapter 3 discussed the effects of user-side censorship typical of state-level censors, where an intermediate device along the path between users and publishers blocks communication.

This study analyzed the consequences of Internet censorship on users, Internet Service Providers (ISPs), and content providers; analyzing network snapshots captured at an ISP in Pakistan over a period of 3 years, beginning in 2011. During this period, the country's censorship policy evolved; specifically, pornographic content was blocked in 2011, and YouTube was blocked in 2012.

Analysis of users' browsing behaviour revealed two patterns. The YouTube block led to a wide adoption of circumvention systems, accompanied by a reduced use of the local ISP's DNS resolvers. The shift to DNS resolvers outside the country has a negative effect on a nation's overall control over its Internet traffic as users transfer their base of trust (i.e. DNS resolution), potentially exposing them to security risks. On the other hand, users of porn content turned to unblocked porn domains, and seemed to be flexible about served content as long as it was within a broad category.

This study also examined the economic impact of censorship. The Supply and Demand model states that in a free and competitive market, the price of a good fluctuates and eventually settles down at a point where the supply of a good exactly matches its demand. But how does this model map to Internet censorship? One way to look at it is that the good is the content served over the Internet, its price is represented by user bandwidth, and the demand is dictated by users' personal preferences in terms of the content that they want to access. When censorship takes place, the supply (content) diminishes, while the demand (user interest) and the price of the good (user bandwidth) remain unchanged; so where does the demand go? Analysis of traffic to popular video and porn providers before and after censorship revealed that a supply shift takes place, directly affecting content providers: user demand for YouTube dropped by half, and shifted to other video content providers who benefited from censorship. In the case of porn traffic, there was a degree of financial loss to the porn industry overall as traffic volume reduced to one-third of its pre-censorship magnitude (even after factoring in traffic to unblocked porn domains).

An implicit cost of censorship is incurred by intermediate actors such as ISPs to comply with the government's demands to enforce censorship. Analysis of the ISP's Web caching behaviour with respect to video content from major content providers revealed that as users moved to encryption-based circumvention mechanisms, the ISP's bandwidth requirement from the upstream provider increased. All video content was primarily fetched from the servers of their respective providers since ISPs cannot in general cache encrypted content; the increased operational cost to ISPs could potentially trickle down to its users by requiring them to pay extra.

6.1.2 Publisher-Side Censorship

The next part of the dissertation shifted to study a new kind of blocking that is mandated by publishers: the user's request arrives at the publisher, but the publisher (or something working on its behalf) refuses to respond based on some property of the user.

Differential Treatment of Tor Users. Chapter 4 employed comprehensive measurements in the context of Tor anonymity network, uncovering significant evidence of Tor blocking: at least 1.3 million IP addresses blocked Tor users at the network layer, and at least 3.67% of Alexa top 1K websites blocked Tor users at the application layer. The websites that blocked Tor mostly belonged to Autonomous Systems (ASes) corresponding to mobile and access ISPs, and hosting services. Some of these ASes performed wholesale blocking of Tor, that is all the IP addresses in the AS blocked Tor. CloudFlare and Akamai stood out as dominant Tor blockers, highlighting the amplified blocking effect that centralized Web services might create when their policy trickles down to thousands of their client websites. Some of this blocking was caused by blacklists that included Tor exit nodes; yet other instances were because of abuse generated from Tor exit nodes, triggering automated blocking mechanisms on websites. The latter observation resonates with the point made in Section 2.7 that the growing trend for CRSes to rely on commercial third-parties to increase the cost of blocking comes with its own limitations: specifically, any change in the third party's policy will affect all CRS users. For example, Meek—a Tor pluggable transport that relays traffic through a server hosted on third-party CDNs such as Google, Azure, and Amazon—recently experienced difficulties when its Google version stopped working. It turned out that rather than being blocked by a censor, it was Google that shut down the Meek server hosted on its cloud platform because of violation of terms of service; a botnet had used Tor and Meek for command and control communication [205].

This study provided a first step towards addressing the problems faced by Tor users by characterizing websites that treat traffic from the Tor network differently from other traffic. The next steps, as described by Tor developer Roger Dingledine [206], involve social activism to engage with major players on the Web such as CloudFlare; getting their perspective on the differential treatment of Tor users, and to discuss possible solutions. There is not much we can do in the case of entities such as ISPs and countries that preemptively block all Tor exit nodes as a matter of policy beyond awareness campaigns to highlight the problem (e.g. Tor's "Don't Block Me" initiative [146]). In the case of abuse-based blocking, we need solutions to enable precise filtering beyond IP address blocking of Tor exit nodes; so that benign Tor users are not forced to bear the consequences of the abusive actions of other Tor users sharing the same exit node (Section 4.5 discusses some possible solutions).

Differential Treatment of Adblock Users. Chapter 5 extended the investigation of publisher-side blocking to users of adblocking software. Online advertising revenue is gravely threatened by the rising popularity of adblockers, prompting publishers to actively detect and block (or warn) adblock users—practices referred to as anti-adblocking. This study employed a novel approach for identifying third-party services shared across multiple websites to present a first characterization of anti-adblocking, revealing that at least 6.7% of Alexa top 5K websites employed anti-adblocking. These practices were

found to be common across a diverse mix of publishers ("General News", "Blogs/Wiki", and "Entertainment" categories), employing 14 unique scripts downloaded from 12 unique domains. Unsurprisingly, the most popular domains were those that have a stake in the game—Google, Taboola, Outbrain, Ensighten and Pagefair (a company that specialises in anti-adblocking services). In some cases, anti-adblocking services were distributed by a domain to client websites belonging to the same organisation: TripAdvisor distributed an anti-adblocking script to its eight websites with different country code top-level domains, and adult websites (all hosted by MindGeek) turned to DoublePimp. Finally, visiting a sample website corresponding to each anti-adblocking script with popular adblockers (AdBlock Plus, Ghostery, and Privacy Badger) revealed that half of the 12 anti-adblocking suppliers were counter-blocked by at least one adblocker; the arms race seemed to have already entered the next level. It is hard to say how many levels deeper the adblocking arms race might go. While anti-adblocking may provide temporary relief to publishers, it is essentially a band-aid solution to mask a deeper issue—the disequilibrium between ads (and, particularly, their behavioural tracking back-end) and information. Any long term solution must address the reasons that brought users to adblockers in the first place.

6.2 Future Directions and Conclusions

This dissertation has has opened up a number of avenues for future work.

Holistic Analysis of Differential Treatment. While user-side censorship has been previously examined in great detail [207], we have only just started to understand differential treatment of users. This dissertation presented two isolated studies in the context of Tor and adblock users, but a broad understanding of the phenomenon is lacking. Specifically, future research should study three aspects of differential treatment: its mechanisms, user perception, and publisher perception. Publisher-side blocking is triggered by some property of the user, but what are these properties? One way to construct a comprehensive list of blocking triggers is to begin with the common suspects: different network types (e.g. academic, VPN, and residential), browsers, countries, and client-side tools that are likely to make users be perceived as 'bad' from the point-of-view of economics and security (e.g. Tor, adblockers, circumvention tools, and open proxies). The next step would be to visit Alexa top 1M websites employing each one of the triggers previously identified, and comparing responses with baseline responses without any trigger. The main challenges will be to distinguish between differential treatment and genuine reasons for inaccessibility (e.g. network artefacts) and application layer customizations (e.g. personalization based on geolocation). While the former can be somewhat mitigated via repeated measurements, the latter will need more consideration. Issues related to censorship are also fundamentally social in character, so computer scientists need to collaborate with social scientists to get a broader context of how differential treatment is perceived by users and publishers. For

example, do users know that it is possible for them to be treated differentially by websites, and how do they decide if that is the case? Does this lead them to modify their browsing behaviour? On publishers' end, it would be useful to gain a deeper understanding of why they choose to block certain classes of users, and to what degree are they willing to relax differential treatment. The effectiveness of such dialogues has been demonstrated by our study in Chapter 4 where CloudFlare was identified as one of the main entities blocking Tor. As a result of media coverage [174], [175] and online debates [176], CloudFlare relented a month after the study was presented by adopting a more permissive policy regarding Tor traffic [177].

Understanding Usability and Usage of Censorship Resistance Systems. Chapter 2 showed that technical aspects of censorship and circumvention have been an active area of research, but we do not fully understand how users bypass censorship by adopting various censorship resistance systems (CRSes). Future studies need to address the following issues. What are the most popular CRSes in different countries? Since a number of CRSes are supported by ads, one way to assess this is to launch an ad campaign targeting users in censored countries and use ad analytics as a proxy for popularity of the CRS ranked by countries. This study could be complemented by qualitative methods (e.g. interviewing developers of CRSes) to understand CRS economic models. CRS usability is another neglected area of study. Future research should develop an evaluation framework to assess CRS usability in terms of performance and user interaction. This could involve benchmarking performances of popular CRSes and conducting usability studies, correlating usability with popularity to understand what makes a CRS popular.

An interesting source of data is keyword searches and public logs (raw or aggregated) about online speech (e.g. Twitter feeds and Google Trends); this data could be correlated with users' intent to circumvent following documented incidents of censorship. This will shed light on the time lag between the occurrence of a censorship incident and people turning to search for means to circumvent. This study will inform CRS developers about how to enable users to find them more effectively.

The Effects of Censorship on Online Advertising. Chapter 3 revealed that after YouTube was blocked in Pakistan, the fraction of encrypted traffic seen at the ISP drastically increased, suggesting increased use of circumvention technologies. Wide adoption of CRSes spurred by state-level censorship could potentially cause economic loss to content providers by undermining targeted advertising. As stated in Section 2.5.3, most practical circumvention tools bypass blocking by relaying traffic through IP addresses in unblocked countries. This makes it difficult for ads to be matched to users based on their geographic location. Additionally, many CRSes also incorporate privacy-preserving features, making it difficult to match ads to users based on their behavioural profiles. So does the aggregate ad revenue generated from a country decline after major censorship events? Getting such

insights is challenging as it depends on finding solutions to hard sub-problems. How important is ad-targeting for online advertising in the first place? How many deployed circumvention tools interfere with targeted advertising, and how do we define mistargeting? Are CRSes that break targeted advertising also the popular ones? Is CRS adoption in censored countries high enough to significantly affect ad-targeting? Another vantage point, even though hard to acquire because of data sensitivity, is historic trends in revenue generation from censored countries observed by key players in the advertising ecosystem: advertisers, publishers, and ad networks.

Characterizing Service Footprint. Studies that aim to study some property of a service first need to identify IPs that offer the service, and then highlight the fraction of this population that exhibits a given property (e.g. X% of the Web has vulnerability Y). However, comprehensive identification of IPs that represent a service is not trivial. The Internet is not a static entity: IP address reachability and service availability varies because of a number of reasons including routing changes, outages, network configurations, and services legitimately going up and down. With the recent advent of fast Internet scanning tools such as ZMap [157], it is becoming more common to identify a service by scanning the entire Internet with TCP SYN probes on the port corresponding to the service (e.g. Section 4.3 employed this technique to highlight the fraction of the Web that blocks Tor by analyzing difference in responses observed from Tor exit nodes and control machines). In theory, this approach seems reasonable and agrees with our mental model of a layered network stack where TCP responses to SYN probes identify alive IPs and SYN-ACK responses establish presence of the target service. However, inference of IP aliveness and service availability based on Internet-wide scans is more nuanced than it seems because of multiple potentially non-overlapping views (e.g. at the granularity of ICMP, IP, TCP, application layer, and application layer semantics), and endemic churn in the availability of IPs and services over time and across different locations. Future studies should outline practical considerations in interpreting Internet-wide scans to study application layer properties, with recommendations for methodological improvements where possible.

Conclusion. To conclude, there is nothing new about censorship: those in power have employed censorship to suppress speech or writings deemed objectionable for as long as human discourse has existed. But with the rise of new networks of communication and information flow facilitated by the digital revolution, censorship can achieve an unprecedented scale while remaining largely invisible; affecting a range of actors beyond the intended targets because of the complex ways in which entities communicate over the Internet and the heterogeneity of the information shared. This dissertation developed methodologies to measure and characterize different aspects of Internet censorship, with an emphasis on its disruptive manifestations (i.e. the censor completely or partially blocks users' access to the target information). However, there exist subtle kinds of

censorship where instead of restricting access to information, the censor actively injects other information on to the primary source to dilute its effect or to bias opinion in its favour. In a recent study, King *et al.* noted that the Chinese government has recruited a large number of people to surreptitiously insert a large volume of comments into social media posts [208]. The goal of these comments is to stop discussions that can lead to collective action by changing the subject to one that is neutral or positive (e.g. cheerleading for China, the revolutionary history of the Communist Party, and other symbols of the regime). Distraction is a more effective strategy to control information than blocking which can be perceived as aggressive, potentially creating furore. Such a subtle form of censorship is also much harder to detect because it requires understanding semantic properties of information.

Because of its continually evolving nature, our understanding of Internet censorship will always remain partial. This dissertation has taken a step towards bringing more transparency to the largely opaque area of Internet censorship. While transparency does not provide a direct answer to censorship, it is a powerful medium to expose surreptitious practices—empowering individuals to make informed choices about their online communications, and facilitating enquiries into the legality and ethics of such practices. What cannot be acknowledged cannot be addressed.

Bibliography

As a large part of the bibliography comprises web-based content, we have included URLs to help readers in locating documents of interest. We validated all these URLs in October 2016, and found them to be working except for two cases (indicated by the note 'Inaccessible')—in these cases, and in case any of the alive links become inoperational in future, we advise users to use a search engine to locate the document's new location, or use a preservation system like `archive.org` to look at the older version of the document when it was working. We have also included year of publication where the URL corresponds to documents of temporal nature (e.g. news articles and blog posts).

[1] U.S. Constitution, "First Amendment." `http://www.law.cornell.edu/constitution/constitution.billofrights.html`, 1791. (Page 12.)

[2] United Nations, "The Universal Declaration of Human Rights." `http://www.un.org/en/universal-declaration-human-rights/index.html`, 1948. (Page 12.)

[3] R. Faris and N. Villeneuve, *Access Denied: The Practice and Policy of Global Internet Filtering*, ch. Measuring Global Internet Filtering, p. 127. Cambridge, MA: MIT Press, 2008. (Page 12.)

[4] W. H. Dutton, A. Dopatka, G. Law, and V. Nash, "Freedom of Connection—Freedom of Expression: The Changing Legal and Regulatory Ecology Shaping the Internet." `http://unesdoc.unesco.org/images/0019/001915/191594e.pdf`, 2011. United Nations Educational, Scientific and Cultural Organization (UNESCO). (Page 12.)

[5] Beacon for Freedom of Expression, "The Long History of Censorship." `http://www.beaconforfreedom.org/liste.html?tid=415&art_id=475`, 2010. (Page 12.)

[6] M. Hilbert and P. Lopez, "The World's Technological Capacity to Store, Communicate, and Compute Information," *Science*, vol. 332, no. 6025, pp. 60–65, 2011. (Page 13.)

[7] R. Deibert, J. Palfrey, R. Rohozinski, and J. Zittrain, *Access Contested: Security, Identity, and Resistance in Asian Cyberspace*, ch. Access Contested: Toward the Fourth Phase of Cyberspace Controls, p. 320. Cambridge, MA: MIT Press, 2011. (Page 13.)

[8] "Global Network Initiative." https://www.globalnetworkinitiative.org. (Pages 14 and 64.)

[9] Freedom House, "Freedom in the World 2016." https://freedomhouse.org/report/freedom-world/freedom-world-2016, 2016. (Page 14.)

[10] gfwrev, "HTTP URL/keyword detection in depth." http://gfwrev.blogspot.jp/2010/03/http-url.html, 2010. (Pages 15 and 27.)

[11] R. Clayton, "Failures in a Hybrid Content Blocking System," in *Proceedings of the Privacy Enhancing Technologies Symposium*, pp. 78–92, Springer, 2006. (Page 25.)

[12] l7-filter. http://l7-filter.sourceforge.net. (Page 25.)

[13] Bro. https://www.bro.org. (Pages 25 and 59.)

[14] Snort. https://www.snort.org. (Page 25.)

[15] nDPI. http://www.ntop.org/products/ndpi/. (Page 25.)

[16] A. Dainotti, A. Pescape, and K. Claffy, "Issues and Future Directions in Traffic Classification," *IEEE Network*, vol. 26, no. 1, pp. 35–40, 2012. (Page 25.)

[17] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in *Proceedings of the Symposium on Security and Privacy*, SP '10, pp. 305–316, IEEE, 2010. (Page 25.)

[18] P. Dorfinger, G. Panholzer, and W. John, "Entropy Estimation for Real-time Encrypted Traffic Identification," in *Proceedings of the International Conference on Traffic Monitoring and Analysis*, pp. 164–171, Springer-Verlag, 2011. (Page 25.)

[19] L. Bernaille and R. Teixeira, "Early Recognition of Encrypted Applications," in *Proceedings of the International Conference on Passive and Active Network Measurement*, pp. 165–175, Springer-Verlag, 2007. (Page 25.)

[20] C. V. Wright, F. Monrose, and G. M. Masson, "On Inferring Application Protocol Behaviors in Encrypted Network Traffic," *Journal of Machine Learning Research*, vol. 7, pp. 2745–2769, 2006. (Page 25.)

[21] B. Wiley, "Dust: A Blocking-Resistant Internet Transport Protocol," tech. rep., School of Information, University of Texas at Austin, 2011. (Pages 25 and 162.)

[22] B. Leidl, "obfuscated-openssh." https://github.com/brl/obfuscated-openssh/blob/master/README.obfuscation, 2009. (Page 25.)

[23] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website Fingerprinting in Onion Routing Based Anonymization Networks," in *Proceedings of the Workshop on Privacy in the Electronic Society*, pp. 103–114, ACM, 2011. (Page 25.)

[24] Q. Sun, D. R. Simon, Y.-M. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu, "Statistical Identification of Encrypted Web Browsing Traffic," in *Proceedings of the Symposium on Security and Privacy*, IEEE, 2002. (Page 25.)

[25] A. Hintz, "Fingerprinting Websites Using Traffic Analysis," in *Proceedings of Privacy Enhancing Technologies Workshop*, Springer-Verlag, 2002. (Page 25.)

[26] G. D. Bissias, M. Liberatore, D. Jensen, and B. N. Levine, "Privacy Vulnerabilities in Encrypted HTTP Streams," in *Proceedings of the International Conference on Privacy Enhancing Technologies*, pp. 1–11, Springer-Verlag, 2006. (Page 25.)

[27] T. Wilde, "Great Firewall Tor Probing." `https://gist.github.com/da3c7a9af01d74cd7de7`, 2015. (Page 25.)

[28] OpenNet Initiative, "China's Green Dam: The Implications of Government Control Encroaching on the Home PC." `https://opennet.net/chinas-green-dam-the-implications-government-control-encroaching-home-pc`, 2009. (Page 26.)

[29] J. Knockel, J. R. Crandall, and J. Saia, "Three Researchers, Five Conjectures: An Empirical Analysis of TOM-Skype Censorship and Surveillance," in *Proceedings of the Workshop on Free and Open Communications on the Internet*, USENIX, 2011. (Page 26.)

[30] 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 Workshop on Privacy in the Electronic Society*, pp. 43–54, ACM, 2012. (Page 26.)

[31] A. Biryukov, I. Pustogarov, and R.-P. Weinmann, "Trawling for Tor Hidden Services: Detection, Measurement, Deanonymization," in *Proceedings of the Symposium on Security and Privacy*, pp. 80–94, IEEE, 2013. (Pages 26 and 52.)

[32] 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 Conference on Computer and Communications Security*, pp. 337–348, ACM, 2013. (Page 26.)

[33] R. Dingledine, N. Hopper, G. Kadianakis, and N. Mathewson, "One Fast Guard for Life (or 9 Months)," in *Workshop on Hot Topics in Privacy Enhancing Technologies*, 2014. (Page 26.)

[34] 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 USENIX Conference on Security*, pp. 227–240, USENIX, 2013. (Page 26.)

[35] CNN, "Opinion: The Chilling Reality of China's Cyberwar on Free Speech." http://www.cnn.com/2015/03/24/opinions/china-internet-dissent-roseann-rife/, 2015. (Page 26.)

[36] D. Bamman, B. O'Connor, and N. Smith, "TAZ Servers and the Rewebber Network: Enabling Anonymous Publishing on the World Wide Web," *First Monday*, vol. 17, no. 3–5, 2015. (Page 26.)

[37] Journalism and Media Studies Centre, "Weiboscope." http://weiboscope.jmsc.hku.hk, 2015. Inaccessible. (Page 26.)

[38] Ars Technica, "The death of SuprNova.org." http://arstechnica.com/staff/2005/12/2153/, 2005. (Page 26.)

[39] Ars Technica, "Massive denial-of-service attack on GitHub tied to Chinese government." http://arstechnica.com/security/2015/03/massive-denial-of-service-attack-on-github-tied-to-chinese-government/, 2015. (Page 26.)

[40] C. Anderson, "Dimming the Internet: Detecting Throttling as a Mechanism of Censorship in Iran." *preprint, arXiv:1306.4361v1 [cs.NI],* http://arxiv.org/pdf/1306.4361v1.pdf, 2013. (Page 27.)

[41] P. Winter and S. Lindskog, "How the Great Firewall of China is Blocking Tor," in *Proceedings of the Workshop on Free and Open Communications on the Internet*, USENIX, 2012. (Page 27.)

[42] Renesys, "Egypt Leaves the Internet." http://research.dyn.com/2011/01/egypt-leaves-the-internet/, 2011. (Page 27.)

[43] Renesys, "The Battle for Tripoli's Internet." http://research.dyn.com/2011/08/the-battle-for-tripolis-intern/, 2011. (Page 27.)

[44] Renesys, "Internet Blackout in Sudan." http://research.dyn.com/2013/09/internet-blackout-sudan/, 2013. (Page 27.)

[45] Renesys, "Myanmar Internet Disruptions." http://research.dyn.com/2013/08/myanmar-internet/, 2013. (Page 27.)

[46] Anonymous, "The Collateral Damage of Internet Censorship by DNS Injection," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 3, pp. 21–27, 2012. (Pages 27 and 57.)

[47] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-generation Onion Router," in *Proceedings of the USENIX Security Symposium*, vol. 13, pp. 21–21, USENIX, 2004. (Pages 30, 44, 45, 88, and 162.)

[48] P. Winter, T. Pulls, and J. Fuss, "ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship," in *Proceedings of the Workshop on Workshop on Privacy in the Electronic Society*, pp. 213–224, ACM, 2013. (Pages 30, 38, 39, 40, 161, and 162.)

[49] M. C. Tschantz, S. Afroz, V. Paxson, and J. D. Tygar, "On Modeling the Costs of Censorship." *preprint, arXiv:1409.3211v1 [cs.CR],* `http://arxiv.org/pdf/1409.3211v1.pdf`, 2014. (Pages 32, 50, and 51.)

[50] 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 Privacy Enhancing Technologies Symposium*, pp. 239–258, Springer, 2012. (Pages 38, 39, 50, 161, and 162.)

[51] 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 Workshop on Free and Open Communications on the Internet*, USENIX, 2012. (Pages 38, 39, and 161.)

[52] The Tor Project, "Tor Bridges Specification." `https://gitweb.torproject.org/torspec.git/tree/attic/bridges-spec.txt`, 2009. (Pages 38 and 39.)

[53] N. Feamster, M. Balazinska, W. Wang, H. Balakrishnan, and D. Karger, "Thwarting Web Censorship with Untrusted Messenger Delivery," in *Proceedings of Privacy Enhancing Technologies workshop*, pp. 125–140, Springer-Verlag, 2003. (Pages 38, 39, and 161.)

[54] E. Y. Vasserman, N. Hopper, and J. Tyra, "SilentKnock : Practical, Provably Undetectable Authentication," *International Journal of Information Security*, vol. 8, no. 2, pp. 121–135, 2009. (Pages 38, 39, and 161.)

[55] D. McCoy, J. A. Morales, and K. Levchenko, "Proximax: Measurement-Driven Proxy Dissemination," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 260–267, Springer, 2012. (Pages 38, 39, and 161.)

[56] M. C. Tschantz, S. Afroz, anonymous, and V. Paxson, "SoK: Towards Grounding Censorship Circumvention in Empiricism," in *Proceedings of the Symposium on Security and Privacy*, pp. 914–933, IEEE, 2016. (Pages 40, 49, and 51.)

[57] B. Laurie and R. Clayton, "Proof-of-Work Proves Not to Work," in *Workshop on Economics and Information Security*, 2004. (Page 41.)

[58] R. Dingledine and N. Mathewson, "Tor Directory Protocol, Version 3." `https://gitweb.torproject.org/torspec.git/tree/dir-spec.txt`, 2006. (Pages 42, 89, and 102.)

[59] H. M. Moghaddam, B. Li, M. Derakhshani, and I. Goldberg, "SkypeMorph: Protocol Obfuscation for Tor Bridges," in *Proceedings of the Conference on Computer and Communications Security*, pp. 97–108, ACM, 2012. (Pages 44, 45, 53, and 162.)

[60] A. Houmansadr, T. Riedl, N. Borisov, and A. Singer, "IP over Voice-over-IP for Censorship Circumvention." *preprint, arXiv:1207.2683v2 [cs.CR],* `https://arxiv.org/pdf/1207.2683v2.pdf`, 2012. (Pages 44, 45, and 162.)

[61] S. Burnett, N. Feamster, and S. Vempala, "Chipping Away at Censorship Firewalls with User-generated Content," in *Proceedings of the 19th USENIX Conference on Security*, pp. 29–29, USENIX, 2010. (Pages 44, 45, 51, and 162.)

[62] S. Khattak, M. Javed, P. D. Anderson, and V. Paxson, "Towards Illuminating a Censorship Monitor's Model to Facilitate Evasion," in *Proceedings of the Workshop on Free and Open Communications on the Internet, FOCI*, USENIX, 2013. (Pages 44, 45, and 162.)

[63] A. Houmansadr, G. T. 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*, pp. 187–200, ACM, 2011. (Pages 44, 45, and 162.)

[64] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A Distributed Anonymous Information Storage and Retrieval System," in *Proceedings of the International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*, pp. 46–66, Springer-Verlag, 2001. (Pages 44, 45, and 162.)

[65] M. Waldman and D. Mazières, "Tangler: A Censorship-resistant Publishing System Based on Document Entanglements," in *Proceedings of the Conference on Computer and Communications Security*, pp. 126–135, ACM, 2001. (Pages 44, 45, 51, 53, and 162.)

[66] J. Geddes, M. Schuchard, and N. Hopper, "Cover Your ACKs: Pitfalls of Covert Channel Censorship Circumvention," in *Proceedings of the Conference on Computer and Communications Security*, pp. 361–372, ACM, 2013. (Pages 47 and 52.)

[67] A. Houmansadr, C. Brubaker, and V. Shmatikov, "The Parrot Is Dead: Observing Unobservable Network Communications," in *Proceedings of the Symposium on Security and Privacy*, pp. 65–79, IEEE, 2013. (Pages 47 and 52.)

[68] 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*, pp. 163–172, ACM, 2014. (Pages 47 and 162.)

[69] Reuters, "Iranians Face New Internet Curbs Before Presidential Election." http://www.reuters.com/article/2013/05/21/net-us-iran-election-internet-idUSBRE94K0ID20130521, 2013. (Page 47.)

[70] bit-smuggler. https://github.com/danoctavian/bit-smuggler. (Pages 47 and 162.)

[71] YOUR-FREEDOM. https://www.your-freedom.net/. (Pages 47 and 162.)

[72] M. Schuchard, J. Geddes, C. Thompson, and N. Hopper, "Routing Around Decoys," in *Proceedings of the Conference on Computer and Communications Security*, pp. 85–96, ACM, 2012. (Page 48.)

[73] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. R. Karger, "Infranet: Circumventing Web Censorship and Surveillance," in *Proceedings of the USENIX Security Symposium*, pp. 247–262, USENIX, 2002. (Pages 48 and 162.)

[74] P. Vines and T. Kohno, "Rook: Using Video Games as a Low-Bandwidth Censorship Resistant Communication Platform," Tech. Rep. UW-CSE-2015-03-03, University of Washington, 2015. (Pages 48 and 162.)

[75] B. Hahn, R. Nithyanand, P. Gill, and R. Johnson, "Games Without Frontiers: Investigating Video Games as a Covert Channel." *preprint, arXiv:1503.05904v2 [cs.CR],* http://arxiv.org/pdf/1503.05904v2.pdf, 2015. (Pages 48 and 162.)

[76] T. Elahi, C. M. Swanson, and I. Goldberg, "Slipping Past the Cordon: A Systematization of Internet Censorship Resistance." CACR Tech Report 2015-10, 2015. (Page 49.)

[77] The Tor Project, "Pluggable Transports." https://www.torproject.org/docs/pluggable-transports.html.en, 2012. (Pages 49 and 50.)

[78] S. Khattak, L. Simon, and S. J. Murdoch, "Systemization of Pluggable Transports for Censorship Resistance." *preprint, arXiv:1412.7448 [cs.CR],* https://arxiv.org/pdf/1412.7448v1.pdf, 2014. (Pages 49, 50, and 51.)

[79] 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*, pp. 47–58, ACM, 2004. (Pages 50 and 161.)

[80] C. S. Leberknight, M. Chiang, H. V. Poor, and F. Wong, "A Taxonomy of Internet Censorship and Anti-censorship." https://www.princeton.edu/~chiangm/anticensorship.pdf, 2012. (Page 50.)

[81] G. Perng, M. K. Reiter, and C. Wang, "Censorship Resistance Revisited," in *Proceedings of the 7th international conference on Information Hiding* (M. Barni, J. Herrera-Joancomart, S. Katzenbeisser, and F. Prez-Gonzlez, eds.), vol. 3727, pp. 62–76, Springer, 2005. (Page 50.)

[82] S. Gardner, "Freedom-supporting technologies: their origins and current state." https://docs.google.com/document/d/1Pa566Vnx9MEuV9gltXOZkypQ3wXyzVyslg6xHuebqqU/, 2016. (Page 50.)

[83] T. Elahi, J. A. Doucette, H. Hosseini, S. J. Murdoch, and I. Goldberg, "A Framework for the Game-theoretic Analysis of Censorship Resistance," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 4, pp. 61–76, 2016. (Pages 50 and 51.)

[84] pt-compose. https://github.com/infinity0/tor-notes/blob/master/pt-compose.rst, 2013. (Page 50.)

[85] "Complete Specification for Generalised PT Composition." https://trac.torproject.org/projects/tor/ticket/10061, 2013. (Page 50.)

[86] J. Massar, I. Mason, L. Briesemeister, and V. Yegneswaran, "JumpBox—A Seamless Browser Proxy for Tor Pluggable Transports," *Security and Privacy in Communication Networks*, p. 116, 2014. (Pages 50 and 162.)

[87] D. Luchaup, K. P. Dyer, S. Jha, T. Ristenpart, and T. Shrimpton, "LibFTE: A Toolkit for Constructing Practical, Format-Abiding Encryption Schemes," in *Proceedings of USENIX Security Symposium*, USENIX, August 2014. (Page 50.)

[88] The Tor Project, "meek." https://trac.torproject.org/projects/tor/wiki/doc/meek, 2014. (Pages 50 and 162.)

[89] Psiphon. https://psiphon.ca/en/index.html. (Page 50.)

[90] The Tor Project, "Fog." https://gitweb.torproject.org/pluggable-transports/fog.git, 2014. (Page 50.)

[91] obfs3. https://gitweb.torproject.org/pluggable-transports/obfsproxy.git/tree/doc/obfs3/obfs3-protocol-spec.txt, 2013. (Pages 50 and 162.)

[92] Adversary Lab. https://github.com/blanu/AdversaryLab/, 2014. (Page 52.)

[93] The Guardian, "History of 5-Eyes Explainer." http://www.theguardian.com/world/2013/dec/02/history-of-5-eyes-explainer, 2013. (Page 52.)

[94] The Tor Project, "How to Handle Millions of New Tor Clients." https://blog.torproject.org/blog/how-to-handle-millions-new-tor-clients, 2013. (Page 52.)

[95] C. Brubaker, A. Houmansadr, and V. Shmatikov, "CloudTransport: Using Cloud Storage for Censorship-Resistant Networking," in *Proceedings of the Privacy Enhancing Technologies Symposium*, Springer, 2014. (Pages 53 and 162.)

[96] "Amazon S3." `https://aws.amazon.com/s3/`. (Page 53.)

[97] "Baidu." `http://www.baidu.com`. (Page 53.)

[98] Google, "Testimony: The Internet in China." `https://googleblog.blogspot.co.uk/2006/02/testimony-internet-in-china.html`, 2006. (Page 53.)

[99] 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. (Page 54.)

[100] The Wall Street Journal, "U.S. Cloud Providers Face Backlash From China's Censors." `http://www.wsj.com/articles/u-s-cloud-providers-face-backlash-from-chinas-censors-1426541126`, 2015. (Page 54.)

[101] S. J. Murdoch and R. Anderson, *Index on Censorship*, vol. 36, ch. Shifting Borders, pp. 156–159. SAGE, 2007. (Page 54.)

[102] Google, "A new approach to China: an update." `https://googleblog.blogspot.co.uk/2010/03/new-approach-to-china-update.html`, 2010. (Page 54.)

[103] Internet Service Providers Association of Pakistan (ISPAK), "`http://www.ispak.pk`." (Page 56.)

[104] Z. Nabi, "The Anatomy of Web Censorship in Pakistan," in *Proceedings of the Workshop on Free and Open Communications on the Internet*, USENIX, 2013. (Pages 56 and 68.)

[105] A. Attaa, "Pakistan Blocks Thousands of Adult Websites." `https://propakistani.pk/2011/10/31/pakistani-isps-start-banning-adult-websites/`, 2011. (Page 56.)

[106] Index on Censorship, "Pakistan: YouTube blocked over anti-Islam film." `https://www.indexoncensorship.org/2012/09/pakistan-youtube-censorship/`, 2012. (Page 56.)

[107] FOX News, "No. 1 Nation in Sexy Web Searches? Call it Pornistan." `http://www.foxnews.com/world/2010/07/12/data-shows-pakistan-googling-pornographic-material.html`, 2010. (Page 56.)

[108] OpenNet Initiative, "Pakistan." `https://opennet.net/research/profiles/pakistan`, 2012. (Page 56.)

[109] Citizen Lab, "O Pakistan, We Stand on Guard for Thee: An Analysis of Canada-based Netsweeper's Role in Pakistan's Censorship Regime." `https://citizenlab.org/2013/06/o-pakistan/`, 2013. (Page 56.)

[110] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson, "Netalyzr: Illuminating the Edge Network," in *Proceedings of the Internet Measurement Conference*, pp. 246–259, ACM, 2010. (Page 56.)

[111] Electronic Frontier Foundation, "Switzerland." `https://www.eff.org/pages/switzerland-network-testing-tool`. (Page 56.)

[112] A. Sfakianakis, E. Athanasopoulos, and S. Ioannidis, "CensMon: A Web Censorship Monitor," in *Proceedings of the Workshop on Free and Open Communications on the Internet*, USENIX, 2011. (Page 56.)

[113] A. Filastò and J. Appelbaum, "OONI: Open Observatory of Network Interference," in *Proceedings of the Workshop on Free and Open Communications on the Internet*, USENIX, 2012. (Page 56.)

[114] J. R. Crandall, E. Barr, D. Zinn, R. East, and M. Byrd, "ConceptDoppler: A Weather Tracker for Internet Censorship," in *Proceedings of the Conference on Computer and Communications Security*, pp. 352–365, ACM, 2007. (Page 56.)

[115] A. M. Espinoza and J. R. Crandall, "Automated Named Entity Extraction for Tracking Censorship of Current Events," in *Proceedings of the Workshop on Free and Open Communications on the Internet*, USENIX, 2011. (Page 56.)

[116] J. Dalek, B. Haselton, H. Noman, A. Senft, M. Crete-Nishihata, P. Gill, and R. J. Deibert, "A Method for Identifying and Confirming the Use of URL Filtering Products for Censorship," in *Proceedings of the Internet Measurement Conference*, pp. 23–30, ACM, 2013. (Page 56.)

[117] S. Aryan, H. Aryan, and J. A. Halderman, "Internet Censorship in Iran: A First Look," in *Proceedings of the Workshop on Free and Open Communications on the Internet*, USENIX, 2013. (Page 56.)

[118] J.-P. Verkamp and M. Gupta, "Inferring Mechanics of Web Censorship Around the World," in *Proceedings of the Workshop on Free and Open Communications on the Internet*, USENIX, 2012. (Page 56.)

[119] 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 Internet Measurement Conference*, pp. 1–18, ACM, 2011. (Page 56.)

[120] N. Weaver, R. Sommer, and V. Paxson, "Detecting Forged TCP Reset Packets," in *Proceedings of the Network and Distributed System Security Symposium*, The Internet Society, 2009. (Page 56.)

[121] S. Alcock and R. Nelson, "Measuring the Impact of the Copyright Amendment Act on New Zealand Residential DSL Users," in *Proceedings of the Internet Measurement Conference*, pp. 551–558, ACM, 2012. (Page 56.)

[122] R. Farahbakhsh, . Cuevas, R. Cuevas, R. Rejaie, M. Kryczka, R. Gonzalez, and N. Crespi, "Investigating the reaction of BitTorrent content publishers to antipiracy actions," in *International Conference on Peer-to-Peer Computing, P2P*, pp. 1–10, IEEE, 2013. (Page 56.)

[123] Citizen Lab, "Routing Gone Wild: Documenting upstream filtering in Oman via India." `https://citizenlab.org/2012/07/routing-gone-wild/`, 2012. (Page 57.)

[124] A. Chaabane, T. Chen, M. Cunche, E. De Cristofaro, A. Friedman, and M. A. Kaafar, "Censorship in the Wild: Analyzing Internet Filtering in Syria," in *Proceedings of the Internet Measurement Conference*, pp. 285–298, ACM, 2014. (Page 57.)

[125] C. Labovitz, "The Other 50% of the Internet." *North American Network Operators' Group (NANOG)*, `https://www.nanog.org/meetings/nanog54/presentations/Tuesday/Labovitz.pdf`, 2012. (Pages 57 and 83.)

[126] M. Bailey, D. Dittrich, E. Kenneally, and D. Maughan, "The Menlo Report," in *Proceedings of the Symposium on Security and Privacy*, vol. 10, pp. 71–75, IEEE, 2012. (Page 63.)

[127] B. Jones, R. Ensafi, N. Feamster, V. Paxson, and N. Weaver, "Ethical Concerns for Censorship Measurement," in *Ethics in Networked Systems Research*, ACM, 2015. (Page 63.)

[128] J. Wright, T. D. Souza, and I. Brown, "Fine-Grained Censorship Mapping: Information Sources, Legality and Ethics," in *Proceedings of the Workshop on Free and Open Communications on the Internet*, USENIX, 2011. (Page 64.)

[129] "Bolo Bhi." `http://bolobhi.org`. (Page 64.)

[130] Security Information Exchange, "`https://www.dnsdb.info/`." (Page 65.)

[131] TeamCymru IP to ASN Mapping. `http://www.team-cymru.org/IP-ASN-mapping.html`. (Page 65.)

[132] McAfee Customer URL Ticketing System. `http://www.trustedsource.org`. (Pages 68, 124, and 126.)

[133] Mozilla Public Suffix List. https://wiki.mozilla.org/Public_Suffix_List. (Page 68.)

[134] H. Duan, N. Weaver, Z. Zhao, M. Hu, J. Liang, J. Jiang, K. Li, and V. Paxson, "Hold-On: Protecting Against On-Path DNS Poisoning," in *Workshop on Securing and Trusting Internet Names*, 2012. (Page 69.)

[135] Alexa, "http://www.alexa.com/topsites." (Pages 70, 79, and 80.)

[136] "Dramas Online." http://dramaonline.com/. (Page 70.)

[137] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube Traffic Characterization: A View from the Edge," in *Proceedings of the Internet Measurement Conference*, pp. 15–28, ACM, 2007. (Page 71.)

[138] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2014-2019 White Paper." http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html, 2014. (Page 73.)

[139] ExtremeTech, "Just how big are porn sites?." http://www.extremetech.com/computing/123929-just-how-big-are-porn-sites, 2012. (Page 74.)

[140] Renesys, "Turkish Internet Censorship Takes a New Turn." http://www.renesys.com/2014/03/turkish-internet-censorship/, 2014. (Page 76.)

[141] S. Crocker, D. Dagon, D. Kaminsky, D. McPherson, and P. Vixie, "Security and Other Technical Concerns Raised by the DNS Filtering Requirements in the PROTECT IP Bill." http://domainincite.com/docs/PROTECT-IP-Technical-Whitepaper-Final.pdf, 2011. (Page 76.)

[142] G. Maier, F. Schneider, and A. Feldmann, "NAT Usage in Residential Broadband Networks," in *Proceedings of the International Conference on Passive and Active Measurement*, pp. 32–41, Springer-Verlag, 2011. (Page 77.)

[143] G. Tyson, Y. Elkhatib, N. Sastry, and S. Uhlig, "Demystifying Porn 2.0: A Look into a Major Adult Video Streaming Website," in *Proceedings of the Internet Measurement Conference*, pp. 417–426, ACM, 2013. (Page 78.)

[144] Techinasia, "Pakistan video site comes up with a way to bypass countrys YouTube block." https://www.techinasia.com/tunepk-site-helps-bypass-youtube-ban-but-only-for-inoffensive-content, 2013. (Page 80.)

[145] Dawn News, "PTCL and Dailymotion join hands in Pakistan." http://www.dawn.com/news/1102464/ptcl-and-dailymotion-join-hands-in-pakistan, 2013. (Page 80.)

[146] The Tor Project, "Don't Block Me." https://trac.torproject.org/projects/tor/wiki/org/projects/DontBlockMe. (Pages 87 and 133.)

[147] R. Dingledine and N. Mathewson, "Design of a Blocking-Resistant Anonymity System," Tech. Rep. 2006-11-001, The Tor Project, 2006. (Pages 87, 89, and 161.)

[148] The Guardian, "Tor: 'The king of high-secure, low-latency anonymity'." http://www.theguardian.com/world/interactive/2013/oct/04/tor-high-secure-internet-anonymity, 2013. (Page 87.)

[149] The Tor Project, "TorMETRICS: Direct users by country." https://metrics.torproject.org/userstats-relay-country.html?graph=userstats-relay-country&end=2015-09-01, 2015. (Page 88.)

[150] The Tor Project, "Tor Pluggable Transport Specification." https://spec.torproject.org/pt-spec, 2015. (Page 89.)

[151] dan.me.uk, "TOR Node List." https://www.dan.me.uk/tornodes. (Page 90.)

[152] sectoor GmbH, "TOR DNSBL - blacklist for Tor servers." http://www.sectoor.de/tor.php. (Page 90.)

[153] The Tor Project, "The public TorDNSEL service." https://www.torproject.org/projects/tordnsel. (Page 90.)

[154] R. Dingledine and J. Appelbaum, "How governments have tried to block Tor." *Chaos Communication Congress*, https://media.torproject.org/outreach-material/presentations/28c3/slides-28c3.pdf, 2012. (Page 90.)

[155] Open Observatory of Network Interference (OONI). https://ooni.torproject.org/. (Pages 91, 109, and 110.)

[156] B. Jones, T.-W. Lee, N. Feamster, and P. Gill, "Automated Detection and Fingerprinting of Censorship Block Pages," in *Proceedings of the Internet Measurement Conference*, pp. 299–304, ACM, 2014. (Page 91.)

[157] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast Internet-wide Scanning and Its Security Applications," in *Proceedings of the USENIX Security Symposium*, pp. 605–620, USENIX, 2013. (Pages 92 and 136.)

[158] R. Padmanabhan, P. Owen, A. Schulman, and N. Spring, "Timeouts: Beware Surprisingly High Delay," in *Proceedings of the Internet Measurement Conference*, pp. 303–316, ACM, 2015. (Page 94.)

[159] Cloudflare support, "Does CloudFlare block Tor?," 2015. https://support.cloudflare.com/hc/en-us/articles/203306930. (Page 102.)

[160] Wikipedia, "Advice to users using Tor." https://en.wikipedia.org/w/index.php?title=WP:TOR&oldid=670864289, 2015. (Page 103.)

[161] Philipp Winter, "exitmap." https://github.com/NullHypothesis/exitmap, 2014. (Pages 103 and 117.)

[162] Damian Johnson, "Stem." https://stem.torproject.org/. (Page 103.)

[163] P. Winter, R. Köwer, M. Mulazzani, M. Huber, S. Schrittwieser, S. Lindskog, and E. Weippl, "Spoiled Onions: Exposing Malicious Tor Exit Relays," in *Proceedings of the Privacy Enhancing Technologies Symposium*, pp. 304–331, Springer, 2014. (Pages 103 and 107.)

[164] R. Dingledine, N. Hopper, G. Kadianakis, and N. Mathewson, "Project: Make it harder to use exits as one-hop proxies," in *Workshop on Hot Topics in Privacy Enhancing Technologies*, 2014. (Page 103.)

[165] The Tor Project, "Onionoo—a Tor network status protocol." https://onionoo.torproject.org/. (Page 108.)

[166] OONI, "HTTP Requests Test." https://github.com/TheTorProject/ooni-spec/blob/6a3c38f2dc/test-specs/ts-003-http-requests.md. (Page 109.)

[167] Citizen Lab, "URL testing lists." https://github.com/citizenlab/test-lists. (Page 109.)

[168] Risks: Things you should know before using ooniprobe. https://ooni.torproject.org/about/risks/. (Page 110.)

[169] R. Henry, K. Henry, and I. Goldberg, "Making a Nymbler Nymble using VERBS," in *Proceedings of the Privacy Enhancing Technologies Symposium*, vol. 6205, pp. 111–129, Springer, 2010. (Page 115.)

[170] P. P. Tsang, A. Kapadia, C. Cornelius, and S. W. Smith, "Nymble: Blocking Misbehaving Users in Anonymizing Networks," *Transactions on Dependable and Secure Computing*, vol. 8, pp. 256–269, Mar. 2011. (Page 115.)

[171] R. Henry and I. Goldberg, "Formalizing Anonymous Blacklisting Systems," in *Proceedings of the Symposium on Security and Privacy*, pp. 81–95, IEEE, 2011. (Page 115.)

[172] R. Jansen, N. Hopper, and Y. Kim, "Recruiting New Tor Relays with BRAIDS," in *Proceedings of the Conference on Computer and Communications Security*, pp. 319–328, ACM, 2010. (Page 116.)

[173] M. Motoyama, K. Levchenko, C. Kanich, D. McCoy, G. M. Voelker, and S. Savage, "Re: CAPTCHAs: Understanding CAPTCHA-solving Services in an Economic Context," in *Proceedings of the USENIX Conference on Security*, pp. 28–28, USENIX, 2010. (Page 116.)

[174] Ars Technica, "Some websites turning law-abiding Tor users into second-class citizens," 2016. http://arstechnica.com/tech-policy/2016/02/some-websites-turning-law-abiding-tor-users-into-second-class-citizens/. (Pages 117 and 135.)

[175] The Register, "Tor users are actively discriminated against by website operators," 2016. http://www.theregister.co.uk/2016/02/25/tor_users_discriminated_against/. (Pages 117 and 135.)

[176] The Tor Project, "Tor bug ticket: Issues with corporate censorship and mass surveillance." https://trac.torproject.org/projects/tor/ticket/18361. (Pages 117 and 135.)

[177] CloudFlare, "The Trouble with Tor," 2016. https://blog.cloudflare.com/the-trouble-with-tor/. (Pages 117 and 135.)

[178] Z. Weinberg, M. Sharif, J. Szurdi, and N. Christin, "Topics of Controversy: An Empirical Analysis of Web Censorship Lists," in *Proceedings of the Privacy Enhancing Technologies Symposium (To appear)*, 2017. (Page 118.)

[179] Mozilla, "Firefox: Most Popular Extensions." https://addons.mozilla.org/en-us/firefox/extensions/?sort=users. (Pages 119 and 124.)

[180] The Guardian, "Major sites including New York Times and BBC hit by 'ransomware' malvertising." https://www.theguardian.com/technology/2016/mar/16/major-sites-new-york-times-bbc-ransomware-malvertising, 2016. (Pages 119 and 128.)

[181] PageFair, "The 2015 Ad Blocking Report." https://blog.pagefair.com/2015/ad-blocking-report/, 2015. (Page 119.)

[182] IAB Tech Lab, "Publisher Ad Blocking Primer." http://www.iab.com/wp-content/uploads/2016/03/IABTechLab_Publisher_AdBlocking_Primer.pdf, 2016. (Page 119.)

[183] The New York Times, "The Ad Blocking Wars." http://nyti.ms/1Qs20YB, 2016. (Page 119.)

[184] M. Z. Rafique, T. Van Goethem, W. Joosen, C. Huygens, and N. Nikiforakis, "It's Free for a Reason: Exploring the Ecosystem of Free Live Streaming Services," in

Proceedings of the Network and Distributed System Security Symposium, The Internet Society, 2016. (Page 120.)

[185] M. Ikram, H. J. Asghar, M. A. Kaafar, B. Krishnamurthy, and A. Mahanti, "Towards Seamless Tracking-Free Web: Improved Detection of Trackers via One-class Learning." *preprint, arXiv:1603.06289 [cs.CR],* `https://arxiv.org/pdf/1603.06289.pdf`, 2016. (Page 120.)

[186] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz, "The Web Never Forgets: Persistent Tracking Mechanisms in the Wild," in *Proceedings of the Conference on Computer and Communications Security*, pp. 674–689, ACM, 2014. (Page 120.)

[187] B. Liu, A. Sheth, U. Weinsberg, J. Chandrashekar, and R. Govindan, "AdReveal: Improving Transparency into Online Targeted Advertising," in *Workshop on Hot Topics in Networks*, 2013. (Page 120.)

[188] Z. Yu, S. Macbeth, K. Modi, and J. M. Pujol, "Tracking the Trackers," in *Proceedings of the International Conference on World Wide Web*, ACM, 2016. (Pages 120 and 128.)

[189] S. Khattak, D. Fifield, S. Afroz, M. Javed, S. Sundaresan, V. Paxson, S. J. Murdoch, and D. McCoy, "Do You See What I See?: Differential Treatment of Anonymous Users," in *Proceedings of the Network and Distributed System Security Symposium*, The Internet Society, 2016. (Page 120.)

[190] J. Mikians, L. Gyarmati, V. Erramilli, and N. Laoutaris, "Crowd-assisted Search for Price Discrimination in e-Commerce: First Results," in *Proceedings of the Conference on Emerging Networking Experiments and Technologies*, pp. 1–6, ACM, 2013. (Page 120.)

[191] A. Hannak, G. Soeller, D. Lazer, A. Mislove, and C. Wilson, "Measuring Price Discrimination and Steering on E-commerce Web Sites," in *Proceedings of Internet Measurement Conference*, pp. 305–318, ACM, 2014. (Page 120.)

[192] L. Chen, A. Mislove, and C. Wilson, "Peeking Beneath the Hood of Uber," in *Proceedings of the Internet Measurement Conference*, pp. 495–508, ACM, 2015. (Page 120.)

[193] A. Hannak, P. Sapiezynski, A. Molavi Kakhki, B. Krishnamurthy, D. Lazer, A. Mislove, and C. Wilson, "Measuring Personalization of Web Search," in *Proceedings of the International Conference on World Wide Web*, pp. 527–538, ACM, 2013. (Page 120.)

[194] X. Xing, W. Meng, D. Doozan, N. Feamster, W. Lee, and A. C. Snoeren, "Exposing Inconsistent Web Search Results with Bobble," in *Proceedings of the International Conference on Passive and Active Measurement*, Springer, 2014. (Page 120.)

[195] C. Bron and J. Kerbosch, "Algorithm 457: Finding All Cliques of an Undirected Graph," *Communications of the ACM*, vol. 16, no. 9, 1973. (Page 121.)

[196] A. Kuhn, S. Ducasse, and T. Gírba, "Semantic Clustering: Identifying Topics in Source Code," *Information Software Technology*, vol. 49, no. 3, pp. 230–243, 2007. (Page 121.)

[197] T. Yamamoto, M. Matsushita, T. Kamiya, and K. Inoue, "Measuring Similarity of Large Software Systems Based on Source Code Correspondence," in *Proceedings of the International Conference on Product Focused Software Process Improvement*, pp. 530–544, Springer-Verlag, 2005. (Page 121.)

[198] M. Falahrastegar, H. Haddadi, S. Uhlig, and R. Mortier, "Tracking Personal Identifiers Across the Web," in *Proceedings of the International Conference on Passive and Active Network Measurement*, pp. 30–41, Springer, 2016. (Pages 124 and 128.)

[199] The Next Web, "This adblocker-blocker helps you get around sites that ban you for hiding ads." `http://thenextweb.com/apps/2016/02/11/around-and-around-we-go/`, 2016. (Page 127.)

[200] Adblock Plus, "Five and oh look, another lawsuit upholds users' rights online." `https://adblockplus.org/blog/five-and-oh-look-another-lawsuit-upholds-users-rights-online`, 2016. (Page 128.)

[201] MEEDIA, "Doppelte Attacke: Springers zwei Fronten-Strategie im Kampf gegen Ad-Blocker." `http://meedia.de/2015/12/15/doppelte-attacke-springers-zwei-fronten-strategie-im-kampf-gegen-ad-blocker/`, 2016. (Page 128.)

[202] The Register, "Ad-blocker blocking websites face legal peril at hands of privacy bods." `http://www.theregister.co.uk/2016/04/23/anti_ad_blockers_face_legal_challenges/`, 2016. (Page 128.)

[203] S. Guha, B. Cheng, and P. Francis, "Privad: Practical Privacy in Online Advertising," in *Proceedings of the Conference on Networked Systems Design and Implementation*, pp. 169–182, USENIX, 2011. (Page 128.)

[204] J. P. Achara, J. Parra-Arnau, and C. Castelluccia, "MyTrackingChoices: Pacifying the Ad-Block War by Enforcing User Privacy Preferences," in *Workshop on the Economics of Information Security*, 2016. (Page 128.)

[205] The Tor Project, "meek-google suspended for terms of service violations (how to set up your own)," 2016. https://lists.torproject.org/pipermail/tor-talk/2016-June/041699.html. (Page 133.)

[206] The Tor Project, "A Call To Arms: Helping Internet Services Accept Anonymous Users." https://blog.torproject.org/blog/call-arms-helping-internet-services-accept-anonymous-users, 2014. (Page 133.)

[207] R. Deibert, J. Palfrey, R. Rohozinski, J. Zittrain, and eds., *Access Denied: The Practice and Policy of Global Internet Filtering*. Cambridge: MIT Press, 2008. (Page 134.)

[208] G. King, J. Pan, and M. E. Roberts, "How the Chinese Government Fabricates Social Media Posts for Strategic Distraction, not Engaged Argument," tech. rep., Harvard University, 2016. Working paper. (Page 137.)

[209] D. Nobori and Y. Shinjo, "VPN Gate: A Volunteer-Organized Public VPN Relay System with Blocking Resistance for Bypassing Government Censorship Firewalls," in *Proceedings of the Symposium on Networked Systems Design and Implementation*, pp. 229–241, USENIX, 2014. (Pages 161 and 162.)

[210] 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*, pp. 93–102, ACM, 2011. (Page 161.)

[211] uProxy. https://www.uproxy.org/. (Pages 161 and 162.)

[212] 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 Conference on Computer and Communications Security*, pp. 109–120, ACM, 2012. (Page 162.)

[213] I. Goldberg and D. Wagner, "TAZ Servers and the Rewebber Network: Enabling Anonymous Publishing on the World Wide Web," *First Monday*, vol. 3, Aug. 1998. (Page 162.)

[214] foe-project, "https://code.google.com/p/foe-project/." (Page 162.)

[215] L. Invernizzi, C. Kruegel, and G. Vigna, "Message in a Bottle: Sailing Past Censorship," in *Proceedings of the Annual Computer Security Applications Conference*, pp. 39–48, ACM, 2013. (Page 162.)

[216] Y. Wang, P. Ji, B. Ye, P. Wang, R. Luo, and H. Yang, "GoHop: Personal VPN to Defend from Censorship," in *International Conference on Advanced Communication Technology*, IEEE, 2014. (Page 162.)

[217] E. Wustrow, S. Wolchok, I. Goldberg, and J. A. Halderman, "Telex: Anticensorship in the Network Infrastructure," in *Proceedings of the USENIX Conference on Security*, pp. 30–30, USENIX, 2011. (Page 162.)

[218] R. J. Anderson, "The Eternity Service," in *Proceedings of the International Conference on Theory and Applications of Cryptology*, pp. 242–252, CTU Publishing House, 1996. (Page 162.)

[219] MailMyWeb, "`http://www.mailmyweb.com`." (Page 162.)

[220] R. Clayton, S. J. Murdoch, and R. N. M. Watson, "Ignoring the Great Firewall of China," in *Proceedings of the Workshop on Privacy Enhancing Technologies*, pp. 20–35, Springer, 2006. (Page 162.)

[221] D. Fifield, G. Nakibly, and D. Boneh, "OSS: Using Online Scanning Services for Censorship Circumvention," in *Proceedings of the Privacy Enhancing Technologies Symposium*, Springer, 2013. (Page 162.)

[222] E. Wustrow, C. M. Swanson, and J. A. Halderman, "TapDance: End-to-Middle Anticensorship Without Flow Blocking," in *Proceedings of the USENIX Security Symposium*, pp. 159–174, USENIX, 2014. (Page 162.)

[223] W. Zhou, A. Houmansadr, M. Caesar, and N. Borisov, "SWEET: Serving the Web by Exploiting Email Tunnels," in *Workshop on Hot Topics in Privacy Enhancing Technologies*, 2013. (Page 162.)

[224] Scholar Zhang (west-chamber-season-1). `https://code.google.com/p/scholarzhang/`, 2010. (Page 162.)

[225] west-chamber-season 2. `https://code.google.com/p/west-chamber-season-2/`, 2010. (Page 162.)

[226] west-chamber-season 3. `https://github.com/liruqi/west-chamber-season-3/`, 2011. (Page 162.)

[227] D. Fifield, C. Lan, R. Hynes, P. Wegmann, and V. Paxson, "Blocking-resistant Communication through Domain Fronting," *Proceedings on Privacy Enhancing Technologies*, vol. 1, no. 2, 2015. (Page 162.)

[228] 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 Workshop on Free and Open Communications on the Internet*, USENIX, 2011. (Page 162.)

[229] M. Waldman, A. Rubin, and L. Cranor, "Publius: A Robust, Tamper-Evident, Censorship-Resistant and Source-Anonymous Web Publishing System," in *Proceedings of the USENIX Security Symposium*, pp. 59–72, USENIX, Aug. 2000. (Page 162.)

[230] J. Holowczak and A. Houmansadr, "CacheBrowser: Bypassing Chinese Censorship without Proxies Using Cached Content," in *Proceedings of the Conference on Computer and Communications Security*, pp. 70–83, ACM, 2015. (Page 162.)

[231] D. Ellard, C. E. Jones, V. Manfredi, W. T. Strayer, B. Thapa, M. V. Welie, and A. W. Jackson, "Rebound: Decoy routing on asymmetric routes via error messages," in *Proceedings of the Conference on Local Computer Networks (LCN)*, pp. 91–99, IEEE, 2015. (Page 162.)

[232] A. Serjantov, "Anonymizing Censorship Resistant Systems," in *International Peer To Peer Systems Workshop*, 2002. (Page 162.)

[233] W. Mazurczyk, P. Szaga, and K. Szczypiorski, "Using Transcoding for Hidden Communication in IP Telephony." *preprint, arXiv:1111.1250 [cs.CR],* `http://arxiv.org/pdf/1111.1250.pdf`, 2011. (Page 162.)

[234] C. Connolly, P. Lincoln, I. Mason, and V. Yegneswaran, "TRIST: Circumventing Censorship with Transcoding-Resistant Image Steganography," in *Proceedings of the Workshop on Free and Open Communications on the Internet*, USENIX, 2014. (Page 162.)

[235] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton, "Protocol Misidentification Made Easy with Format-Transforming Encryption," in *Proceedings of the Conference on Computer and Communications Security*, ACM, 2013. (Page 162.)

[236] 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 Workshop on Free and Open Communications on the Internet*, USENIX, 2014. (Page 162.)

[237] K. P. Dyer, S. E. Coull, and T. Shrimpton, "Marionette: A Programmable Network Traffic Obfuscation System," in *Proceedings of the USENIX Security Symposium*, pp. 367–382, USENIX, 2015. (Page 162.)

[238] S. Cao, L. He, Z. Li, and Y. Yang, "SkyF2F: Censorship Resistant via Skype Overlay Network," in *Proceedings of the International Conference on Information Engineering*, vol. 1, pp. 350–354, IEEE, 2009. (Page 162.)

[239] T. Ruffing, J. Schneider, and A. Kate, "Identity-Based Steganography and Its Applications to Censorship Resistance." Workshop on Hot Topics in Privacy Enhancing Technologies, 2013. (Page 162.)

[240] The Tor Project, "Tor: Hidden Service Protocol." https://www.torproject.org/docs/hidden-services.html.en. (Page 162.)

[241] Message Stream Encryption. http://wiki.vuze.com/w/Message_Stream_Encryption. (Page 162.)

[242] AnchorFree. http://www.anchorfree.com/. (Page 162.)

[243] GTunnel. http://gardennetworks.org/products. Inaccessible. (Page 162.)

[244] obfs2. https://gitweb.torproject.org/pluggable-transports/obfsproxy.git/tree/doc/obfs2/obfs2-protocol-spec.txt. (Page 162.)

[245] JAP Anonymity & Privacy. https://anon.inf.tu-dresden.de/index.html. (Page 162.)

[246] Lantern. https://getlantern.org/. (Page 162.)

[247] obfs4. https://github.com/Yawning/obfs4/blob/5bdc376e2abaf5ac87816b763f5b26e314ee9536/doc/obfs4-spec.txt. (Page 162.)

[248] Q. Wang, X. Gong, G. T. K. Nguyen, A. Houmansadr, and N. Borisov, "Censor-Spoofer: Asymmetric Communication using IP Spoofing for Censorship-Resistant Web Browsing," in *Proceedings of the Conference on Computer and Communications Security*, ACM, 2012. (Page 162.)

[249] CGIProxy. https://www.jmarshall.com/tools/cgiproxy/. (Page 162.)

[250] Ultrasurf, "http://ultrasurf.us." (Page 162.)

[251] Freegate. http://dit-inc.us/freegate.html. (Page 162.)

Appendix A

Surveyed Censorship Resistance Systems

Table A.1: Surveyed systems relevant to different schemes of Communication Establishment (Section 2.4). † Academic paper, ⋆ Deployed.

High Churn Access	Rate Limited			Active Probing		Trust-Based Access
	Proof of Life/Work	Time Partitioning	Keyspace Partitioning	Obfuscating Aliveness	Obfuscating Service	
Flashproxy [50]†⋆	Defiance [51]†	Tor Bridges [147]⋆	Keyspace-Hopping [53]†	SilentKnock [54]†⋆	ScrambleSuit [48]†⋆	Proximax [55]†
VPN-Gate [209]†⋆	Köpsell *et al.* [79]†			BridgeSPA [210]†⋆		uProxy [211]⋆

Table A.2: Surveyed systems relevant to different schemes of Conversation (Section 2.5). † Academic paper, ⋆ Deployed.

Access-Centric Schemes						Publication-Centric Schemes	
Content/Flow Obfuscation				Destination Obfuscation		Content	Distributed
Mimicry	Tunnelling	Covert Channel	Traffic Manip.	Proxy	Decoy Routing	Redundancy	Storage
StegoTorus [212]†	Freewave [60]†	Collage [61]†	Khattak *et al.* [62]†	Flashproxy [50]†⋆	Cirripede [63]†	Rewebber [213]†	Tangler [65]†
FOE [214]†	Facet [68]†	MIAB [215]†	GoHop [216]†⋆	VPN-Gate [209]†⋆	Telex [217]†	Eternity [218]†	
MailMyWeb [219]⋆	JumpBox [86]†	Infranet [73]†⋆	Clayton *et al.* [220]†	OSS [221]†	TapDance [222]†	Freenet [64]†⋆	
SWEET [223]†	CloudTransport [95]†	Castle [75]†	Zhang *et al.* [224] [225] [226]⋆	Domain Fronting [227] [88]†⋆	Curveball [228]†	Publius [229]†	
SkypeMorph [59]†	Castle [75]†	Rook [74]†	CacheBrowser [230]†⋆	Freewave [60]†	Rebound [231]†	Serjantov [232]†	
TransTeg [233]†	Rook [74]†	TRIST [234]†		Rewebber [213]†			
FTE [235]†⋆	Bit-Smuggler [70]⋆	Facade [236]†		Tor [47]†⋆			
Marionette [237]†	SkyF2F [238]†	IBS [239]†		Tor Hidden Services [240]⋆			
ScrambleSuit [48]†⋆	YourFreedom [71]⋆			YourFreedom [71]⋆			
MSE [241]⋆				AnchorFree [242]⋆			
Dust [21]†				GTunnel [243]⋆			
obfs2 [244]⋆				JAP [245]⋆			
obfs3 [91]⋆				Lantern [246]⋆			
obfs4 [247]⋆				uProxy [211]⋆			
CensorSpoofer [248]†				CGIProxy [249]⋆			
				Ultrasurf [250]⋆			
				Freegate [251]⋆			
				CensorSpoofer [248]†			

Appendix B

A Survey of User Perceptions in Pakistan of Internet Censorship

Table B.1: The results of an online survey targeting users in Pakistan to understand their perceptions about the porn block (2011), the YouTube block (2012), and Internet censorship in general (Section 3.2.5). Information about the opportunity to take the survey was disseminated through mailing lists and classroom discussions in Pakistan. We received 770 responses, 75% from male participants and 25% from female participants. Because of the method of survey dissemination, participants were mostly young university students in Computer Science and Engineering departments. Note that the makeup of our survey participants does not reflect the real demographics of the broader population of Pakistan. The results of this survey are not intended to be representative, but rather as illuminating some aspects of how censorship affects Pakistani users.

How do you access videos on the Internet (e.g. music, tv shows/dramas etc)?	
Through Youtube. (I know a method to unblock Youtube in Pakistan).	61.8%
Through another site (vimeo, dailymotion, vidpk.com, etc.)	56.8%
I just search for content on a search engine and click on a link other than Youtube in search results	31.8%
Other	6.7%
I have no interest in accessing videos on the Internet	3.1%
Someone you know sent you a link to a popular YouTube video (in email or via chat, for example). What is the most likely thing you would do?	
I will use a proxy/some other method and try to watch the video on YouTube	52.6%
I will ignore the link. It's a hassle to use Youtube unblocking methods to watch just one video	36.0%
I will search for that video on another video sharing website (vimeo, dailymotion, etc.)	23.3%
I will ignore the link. YouTube is inaccessible in Pakistan.	19.6%
Other	4.7%

Continued on next page

What website, alternate to YouTube, would you recommend to a friend for watching videos in Pakistan?	
Dailymotion	63.8%
Vimeo	41.7%
Tune.pk	23.3%
I don't know any other video sharing website	12.4%
Other	9.2%
vidpk.com	4.0%
FriendsKorner.com	1.7%
What proxy website/VPN service would you recommend to a friend for unblocking YouTube in Pakistan?	
Hotspot shield	51.1%
Other	26.0%
Tor	17.7%
Ultrasurf	15.2%
I don't know any proxy website/ VPN service	12.8%
Open VPN	12.3%
youtubebeatschool.info	1.5%
If Facebook were to be blocked in Pakistan today, how would you learn how to access it?	
Search on the Internet	74.9%
Ask a friend/relative	39.0%
Look on a specific website/blog that I regularly read	17.7%
Call/email a professional technical person	4.5%
Other than YouTube, have you ever been denied access to gmail or google drive or another google service in recent months?	
No, never	67.0%
Yes, but rarely	20.9%
Yes, sometimes. Couple of times a month	6.2%
Yes, often. Couple of times a week or more	2.4%
Other	1.8%
I don't use any Google services	1.7%

Continued on next page

Did you ever upload videos on YouTube before it was blocked?	
I never uploaded anything on YouTube	67.1%
Yes, I did. But now, I upload elsewhere and will come back to YouTube once it is unblocked	21.4%
Yes, I did. And I still upload on YouTube	5.2%
Other	3.3%
Yes, I did. But now, I upload elsewhere and will likely not come back to YouTube	3.0%
Before YouTube was blocked, what devices did you use for watching YouTube videos	
My home computer/personal laptop	96.8%
My cellphone	46.2%
My office/university/college computer	41.0%
My tablet	23.2%
Other	1.6%
After the blocking of YouTube, I still use YouTube on	
My home computer/ personal laptop	84.3%
My office/university/college computer	17.5%
My cellphone	15.5%
Other	12.5%
My tablet	7.2%
The YouTube unblocking method that I figured out:	
On my home computer/personal laptop is inconvenient	39.7%
All unblocking methods I use are such a hassle	35.8%
On my cellphone is inconvenient	26.3%
All unblocking methods I use are easy-to-use.	21.0%
On my office/university/college computer is inconvenient	15.3%
On my tablet is inconvenient	11.9%
Other	5.8%
Do you need Youtube back?	
Yes, although I have learnt how to unblock	71.4%
Yes, I haven't learnt how to unblock	8.1%
Other	7.7%
Doesn't matter. I have found an alternative video sharing websites that are equally good	7.3%
Doesn't matter. I have learnt how to unblock Youtube	5.6%

Continued on next page

Top reason why it is okay to do Internet censorship in Pakistan?	
There are too many websites in the wild that must not be shown to the children	30.6%
Other	25.9%
To block anti-religious content that people put out there	24.2%
People watch too much porn	19.3%

Top reason why it is okay to do Internet censorship in Pakistan?	
I can decide myself what websites/videos to visit. Why should government decide?	45.0%
I want access to youtube educational videos	37.4%
I want access to youtube entertainment videos (movies, drama, cricket, etc.)	9.3%
Other	8.3%

Glossary

Adblocker	(Also Ad-Blocker or Ad Blocker.) Software, typically available as browser plugin, that offers Adblocking.
Adblocking	(Also Ad-Blocking or Ad Blocking) Practice of removing ads from Web documents.
Anti-Adblocking	Publisher-side practice of developing and deploying mechanisms for detecting or counter-blocking adblockers.
AS	Autonomous System; A network or a collection of networks owned by a single entity.
ASN	Autonomous System Number; Globally unique identifier associated with an AS.
BGP	Border Gateway Protocol; protocol for routing IP traffic among ASes.
BitTorrent	p2p protocol for sharing data and files over the Internet.
BRAS	Broadband Remote Access Server; routes traffic between ISP subscribers (access network) and the ISP's core network.
Browser	Software application for viewing information on the Web.
Browser Plugin	Piece of software that enhances and customizes the functionality of standard browsers.
Cache Server	System or application that locally saves popular Web pages and other content to reduce bandwidth demands and for faster access.

CAPTCHA	(Backronym for "Completely Automated Public Turing test to tell Computers and Humans Apart".) Challenge-response test that is used to distinguish humans from non-human actors such as automated scripts.
CDN	Distributed network of proxy servers for efficient delivery of Web resources to users based on geographic locations of the user and the requested content.
Cyberspace	The notional environment enabled by the Internet where communication takes place digitally.
DHCP	Dynamic Host Configuration Protocol; protocol by which a central server automatically provides IP addresses and other configuration data to systems in the network.
Directory Authority	Service that maintains a list of available and trusted Tor nodes.
Directory Consensus	List of available and trusted Tor nodes that is verified by Tor directory authorities (currently nine, which are hardcoded into Tor client). This list is made available to clients directly and through other Tor nodes.
DNS	Domain Name Service; hierarchical, distributed database that stores information about participating systems, mainly domain name to IP address mapping.
DNS Resolver	Server that receives DNS requests from users (typically to map a given domain name to an IP address), and provides an answer directly or by asking other servers.
Entry Node	First hop in Tor that receives user traffic.
Ethernet	Family of network technologies that describe how systems should connect in LANs.
Exit Node	Last hop in Tor that removes the innermost layer of encryption, and sends original data to the destination.
Exit Policy	Policy configured by Tor exit nodes to describe the IP addresses and ports to which the node is willing to carry traffic.
FTP	File Transfer Protocol; protocol for transferring files between a client and server.

Hash	Output of a cryptographic hash function that takes variable size input and outputs a short string (the hash) that uniquely corresponds to the input, and thus serves to authenticate it.
HTML	HyperText Markup Language; standard language for writing Web pages.
HTTP	HyperText Transfer Protocol; underlying protocol used by the Web for transferring documents.
HTTPS	HTTP Secure; HTTP carried over a protocol that uses cryptography to secure communication, such as TLS or SSL.
ICMP	Internet Control Message Protocol; supporting protocol for reporting errors and diagnostics in IP communication.
IP	Internet Protocol; principal protocol that enables communication over the Internet.
IP address	IP address; numerical label that identifies systems in IP networks.
IPv4	Version 4 of IP; uses 32 bit addressing scheme.
IPv6	Recent version 6 of IP; uses 128 bit addressing scheme to deal with exhaustion of IPv4 addresses.
IRC	Internet Relay Chat; protocol that supports text-based distributed real-time conversations (chats).
ISP	Internet Service Provider; entity that provides users access to the Internet, usually in addition to other services.
JavaScript	Scripting language to program the Web.
LAN	Local Area Network; locally managed network that connects systems within a limited area, such as a home or office.
Middle Node	Hop that relays traffic within Tor; it is neither the entry nor the exit point of traffic.
NAT	Network Address Translation; method for translating IP addresses between different IP networks to reroute traffic without readdressing systems.
Online Advertisement	(Shortened to Advert or simply Ad.) Promotional material that is delivered over the Internet .

Online Advertising	Strategy for using the Internet to deliver marketing messages to customers. Typically involves a publisher that embeds the ad in its online content, and an advertiser that provides the ad to be displayed. Between these two, there could be other intermediaries that track statistics and match ads to users.
p2p	(peer-to-peer.) Distributed system that connects and distributes tasks between equally privileged nodes called peers.
pcap	(packet capture.) Mechanism implemented by operating system libraries for capturing network traffic Also refers to the file format used to store traffic.
Pluggable Transport	System that can interface with Tor in a plug-and-play fashion to enable censorship evasion.
Port	Communication endpoint in an operating system that, together with an IP address and protocol type (TCP or UDP), uniquely identifies a service over the Internet.
Protocol	In computer networks, refers to a set of rules that determines how communication should take place between two or more entities.
Proxy	Intermediary system or application that exchanges traffic between clients and servers.
Redirection	Technique for moving users to a different destination than the one originally requested.
Routing	Process of sending traffic within a network or across multiple networks so that it reaches appropriate destinations.
SOCKS	Protocol that routes traffic between a client and server through a proxy server.
Spam	Unsolicited email, often sent for commercial purposes or to spread malware.
SSL	Secure Sockets Layer. (Predecessor of TLS.) Protocol that uses cryptography to secure network communication.
Targeted Advertising	Form of online advertising that matches ads to users based on different properties, such as their geographic location, gender, age, and browsing history.

TCP	Transmission Control Protocol; reliable stream-oriented protocol to deliver traffic over IP networks. A number of major Internet applications are layered over TCP.
TLS	Transport Layer Security. (Successor of SSL.) Protocol that uses cryptography to secure network communication.
Tor	Software for anonymous communication. It encrypts data (including destination IP addresses) multiple times, and relays it over a virtual path with many hops. Each hop decrypts data once to reveal IP address of the next hop. The last hop removes the innermost layer of encryption, and sends original data to the destination. Thus any single hop only knows about its predecessor and successor. The hop that receives original data from users is called entry node (some entry nodes called bridge nodes are not publicly disclosed to resist censorship). The last hop from which traffic leaves the Tor network is called exit node. The other hops are called middle nodes. DNS traffic is handled in a similar fashion, that is the exit node resolves queries on behalf of the user.
Tracker	Script that provides statistics about user behaviour on a website to the publisher or a third-party.
TTL	Time To Live; number of hops over which an IP packet may be forwarded before it expires and is discarded.
Tunneling	Encapsulating one protocol in another such that a user can access network services that are not directly supported by the underlying network, for example carrying IPv6 over IPv4.
URL	Uniform Resource Locator; identifier for Web resources that includes information about its location on the Internet and on the server where it is stored.
VoIP	Voice over IP; methodology for carrying voice communications and multimedia sessions over IP networks such as the Internet.

VPN	Virtual Private Network; extends a private network across the Internet such that remote users can communicate with the private network as if they were physically connected to it.
Web Page	Document that is suitable for being viewed over the Web using a Web browser.
Web Server	System that serves Web documents over HTTP.
WWW	The World Wide Web (abbreviated WWW or the Web); a system for information management where users can access resources by their identifiers (URLs). These resources are interlinked by hypertext links.