Master's thesis

Master's Programme in Computer Science

Governmental Internet censorship and its circumvention: Case of the Great Firewall of China

Akseli Vierimaa

May 11, 2025

Faculty of Science

University of Helsinki

Contact information

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki,Finland

Email address: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Study programme
Faculty of Science	Master's Programme in Computer Science

Tekijä — Författare — Author

Akseli Vierimaa

Työn nimi — Arbetets titel — Title

Governmental Internet censorship and its circumvention: Case of the Great Firewall of China

Ohjaajat — Handledare — Supervisors

Markku Kojo, Valtteri Niemi

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's thesis	May 11, 2025	69 pages

Tiivistelmä — Referat — Abstract

Governmental Internet censorship can be defined as the practice of blocking access to information on the Internet instigated by a country's leadership. While most countries practice Internet censorship, the degree of it varies greatly by country. In many authoritarian countries Internet censorship is used to suppress opposing political views and restrict access to objective information, which makes them especially interesting targets for Internet censorship research.

China is one of the most aggressive censors in the world, and the country operates arguably the world's most elaborate Internet censorship systems. The system in China's main Internet infrastructure that filters packets travelling between China and the outside world is known as the Great Firewall. The Great Firewall is one of the oldest and most well-known packet filtering systems in the world, and it has been the target of active research since its birth. The Great Firewall censors web traffic crossing China's digital border mainly by disrupting the normal flow of DNS queries, and HTTP and HTTPS requests. The Great Firewall also has components that target specific protocols and tools that help users circumvent governmental Internet censorship. Together these components form a comprehensive and dynamic system that has been effective in separating China from the global Internet.

In this thesis we provide an overview on governmental Internet censorship with a strong focus on functionality of the Great Firewall of China. We discuss the functionalities of different components of the Great Firewall and describe what results regarding their effectiveness and extent of censorship have been uncovered. Through this literature review we aim to provide an overview of how the Great Firewall functions, what methods exist to circumvent it, as well as what role the different censorship and circumvention methods play both in China and in the global Internet censorship landscape.

ACM Computing Classification System (CCS)
General and reference → Document types → Surveys and overviews
Social and professional topics → Computing / technology policy → Censorship

Avainsanat — Nyckelord — Keywords

Internet censorship, the Great Firewall of China, Internet censorship circumvention

Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki University Library

Muita tietoja — övriga uppgifter — Additional information

Networking study track

Contents

1 Introduction **1**

2 Internet censorship **3**

2.1 Basics of Internet censorship 3

2.2 Common Internet censorship methods 5

2.3 Studying packet filtering systems 15

2.4 Internet censorship landscape in China 17

3 DNS censorship in the Great Firewall **20**

3.1 DNS censorship 20

3.2 DNS censorship by resolvers 25

3.3 Circumventing DNS censorship: Encrypted DNS 27

4 Web traffic censorship in the Great Firewall **30**

4.1 HTTP censorship 30

4.2 HTTPS censorship 32

4.3 Circumventing web traffic censorship: Encrypted Server Name Indication . 35

4.4 Circumventing web traffic censorship: TCP- and TLS-based methods . . . 38

5 Censorship of circumvention tools in the Great Firewall **42**

5.1 Generic circumvention tools and their censorship 43

5.2 Passive analysis and active probing 47

5.3 Fully passive censorship of circumvention tools 50

5.4 Censorship resistant circumvention 52

6 Discussion **54**

7 Conclusions **57**

References **59**

1 Introduction

Internet censorship can be defined as the practice of removing, manipulating, or blocking access to information on the Internet. Internet censorship overall ranges from companies regulating what is allowed within their company's digital premises to large-scale censorship orchestrated by governments. Censorship usually targets a specific group such as citizens of a country, people participating in a social movement or users of a digital service. Censorship is used as a tool to control what the targeted group is able to access and do on the Internet.

Governmental Internet censorship is Internet censorship that is in some way instigated by a country's leadership. Governmental Internet censorship is large-scale censorship that affects nearly all residents of a country and is used for national control of the Internet.

A decrease in Internet freedom has been an ongoing trend around the world for many years now [77] [41] [61]. Many countries practice some level of Internet censorship [54] [61], but the severity and primary targets of censorship vary by country as does the associated legislation. Internet censorship is mostly guided by national laws. In western countries Internet censorship has been used for example to restrict access to child abuse material or sites infringing on copyright laws: In Finland there is a law that allows the National Bureau of Investigation (Finnish: Keskusrikospoliisi) to maintain a secret list of sites reported to contain child abuse material that Internet Service Providers (ISPs) can choose to censor for their customers [1]. As another example, the USA has extensive copyright legislation that is used to back up requests for take downs of infringing content online [81].

Internet censorship can also be used by governments for political control. It has been shown that aggressive governmental Internet censorship is connected to authoritarianism [41] [25] [29] [13], where Internet censorship is used to suppress opposing political views and restrict access to objective critical information. The main disadvantages of digital authoritarian practices to citizens are increased unreasonable surveillance, restrictions to freedom of expression, prevalence of disinformation, and increased obstacles to critical information [29]. All of this makes authoritarian countries especially interesting targets for Internet censorship research.

China has arguably the world's most elaborate Internet censorship systems. The system in China's main Internet infrastructure that filters packets travelling between China and the

outside world is known as the Great Firewall (GFW). The Great Firewall is a black box to largely everyone else except its operators, but being one of the oldest and most well-known packet filtering systems in the world, it has been the target of active research since its birth. Since it is possible to probe the system from both outside and inside of China, it is possible to infer information about how the system works and how it is evolving.

The Great Firewall censors web traffic crossing China's digital border mainly by disrupting the normal flow of Domain Name System (DNS) requests [47] [9], and Hypertext Transfer Protocol (HTTP) and Hypertext Transfer Protocol Secure (HTTPS) requests [70] [46] [20]. The Great Firewall also has components that target specific protocols and services whose aim is to help users circumvent the above mentioned web censorship [89] [6] [91]. Together these components form a comprehensive and dynamic system that has been effective in separating China from the global Internet.

In this thesis we provide an overview on governmental Internet censorship with a strong focus on functionality of the Great Firewall of China. We discuss the functionalities of different components of the Great Firewall and describe what results regarding their effectiveness and extent of censorship have been uncovered. Through this literature review we aim to provide an overview of how the Great Firewall functions, what methods exist to circumvent it, as well as what role the different censorship and circumvention methods play both in China and in the global Internet censorship landscape. While the censorship methods used in the Great Firewall have also been implemented in other countries and the related circumvention techniques can be used elsewhere as well, the focus of this thesis is on how the methods have been implemented in China specifically.

The rest of the thesis is structured as follows: In Chapter 2 we discuss the basic concepts in Internet censorship and introduce the most common methods used by governments to censor the Internet. We also introduce a basic research setup commonly used in Internet censorship research targeting large national packet filtering systems. In the following three chapters we take a deeper look into the different functionalities of the Great Firewall. In Chapter 3 we describe the DNS censoring capabilities of the Great Firewall and discuss related DNS specific circumvention methods. Chapter 4 is dedicated to censorship of HTTP and HTTPS and circumvention methods that are specific to the two protocols. In Chapter 5 we discuss the workings of generic Internet censorship circumvention protocols developed to circumvent Internet censorship of any kind as well as how these protocols are being censored in China. In Chapter 6 we discuss challenges related to Internet censorship research. Chapter 7 concludes the thesis.

2 Internet censorship

In this chapter we introduce the basic concepts of Internet censorship. Our focus is on the protocols that are under censorship in China through the Great Firewall. In Section 2.1 we introduce basic concepts in Internet censorship and explain common and useful terms. In Section 2.2 we revise DNS, HTTP and HTTPS protocols from Internet censorship point of view and explain how each of the protocols is commonly censored. In addition, we take a look at how the methods compare in terms of global popularity. In Section 2.3 we discuss how packet filtering systems such as the Great Firewall are commonly studied. In order to better understand the Great Firewall specific research results in chapters 3 through 5, it is useful to be familiar with the context in which the censorship takes place. For this purpose we briefly introduce the Internet landscape in China from an Internet censorship point of view in Section 2.4.

2.1 Basics of Internet censorship

In the context of Internet censorship, **a censorship device** is a device in the network whose purpose is to block access to specified resources or services on the Internet. A censorship device is commonly referred to as **a censor** and the words are used interchangeably. It is customary to also refer to the humans who operate these devices as censors, but to avoid confusion, when we need to specifically refer to the goals and motivations of the humans operating the censorship devices, we refer to them as operators. The operators of the censorship system use censorship devices to automate the censoring process. The operators dictate which resources or services should be blocked and configure the censorship devices accordingly.

In an Internet censorship scenario we have a user, a resource on the Internet the user is trying to access, and a censor who is configured to block access to the resource. This basic setup is depicted in Figure 2.1. The resource, for example a web page, is hosted on a server. The user typically uses a client software, such as a web browser, to access the resource on the server. In order to block the user's access to the resource, the censor needs to be located in the network sense between the client and the server. More specifically, the user's request for the target resource needs to pass through the censor in order for the

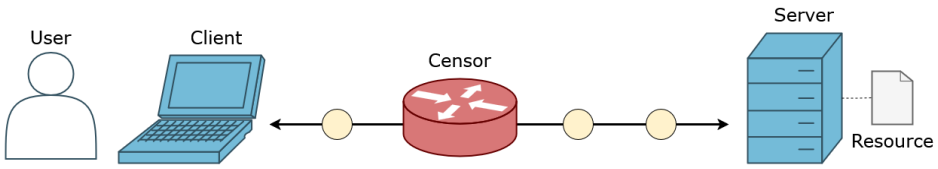

Figure 2.1: Internet censorship scenario. The censor is in between the client and the server and inspects the packets passing through. The yellow circles represent other nodes in the network.

censor to inspect and subsequently interfere with the request.

A censor observes the traffic passing through it, and in case it notices traffic towards a resource categorised as unwanted it moves to block the connection between the client and the server. How the censor detects unwanted traffic is dependent on the protocol carrying the traffic. A censor can block a connection by dropping the unwanted packets or by crafting and injecting packets to the client and the server that cause them to stop communicating with each other.

Censorship devices can be categorised into two categories based on their characteristics. An in-path censor is a networking device that does packet forwarding as its main functionality but also has censorship capabilities [84] [54] [17] [62]. An in-path censor can be for example a router that has software with censorship capabilities installed and configured on it. Since in-path censors have full access to the packets passing through them, they can manipulate the contents of the original packets or completely drop them, as well as inject crafted packets to the connection between the client and the server. Because an in-path censor does censoring in addition to its normal networking operations such as packet forwarding, in-path censors tend to operate under strict time constrains and usually cannot do time-consuming analysis on the packets.

A censor is said to be on-path if it is not an actual node on the network path from the client to the server, but instead it only inspects packets passing through the point where it is connected to the path [84] [54] [17] [62]. An on-path censor is typically a specialized middlebox that can do time-consuming packet analysis since it has no other purpose except to censor traffic. An on-path censor cannot manipulate or drop the original packets, it can only interfere with a connection by injecting new crafted packets. Most components of the GFW are on-path censors.

Governmental Internet censorship devices overall tend to reside close to the censoring nation's border routers, measured by hops, and be deployed in high ranking Autonomous

Systems (ASes) and in large ISP networks with many customers [54] [69] [8].

In Internet censorship **collateral damage** is an important concept that refers to content that is blocked as a consequence of censorship but which the operators of the censorship system did not intend for the censorship device to block. An example would be a scenario where a censor is blocking all websites that contain a predefined keyword in their domain name. In this case innocent sites whose domain names coincidentally contain this keyword would be censored as collateral damage. Usually the operators are vary of causing collateral damage with their censorship system and censorship circumvention techniques often try to leverage this by making it too expensive for censors to block something due to excessive collateral damage.

2.2 Common Internet censorship methods

Web traffic is the number one target of Internet censorship. This means when a government wants to censor something on the Internet, they aim their censorship at the protocols that form the core of web browsing: DNS, HTTP or HTTPS. When a user types in an URL to their browser or clicks a link that takes them to a website, the browser first makes a DNS query to solve the domain name to its IP address. This is the first chance for the censors to interfere. The act of manipulating DNS responses with the goal of preventing the client from receiving the correct IP address is called DNS censorship. After the client has solved the domain name to its IP address, the browser makes an HTTP or HTTPS request towards the server the IP address belongs to. This is the second chance for the censors to interfere. The act of interfering with the normal HTTP or HTTPS request-response cycle is called HTTP or HTTPS censorship respectively.

In this section we discuss how DNS, HTTP and HTTPS censorship work. We concentrate on these methods specifically due to their prominence in the global Internet censorship landscape as well as due to how DNS, HTTP and HTTPS censorship components form the core functionality of the Great Firewall of China. While we cover these methods separately, it is good to note that countries can utilize multiple different censorship methods at the same time, meaning a single user can be under DNS, HTTP, HTTPS and other forms of censorship simultaneously.

Before we explain how DNS, HTTP, and HTTPS censorship work, it is necessary to understand how censors can severe a TCP connection through **a TCP reset attack**.

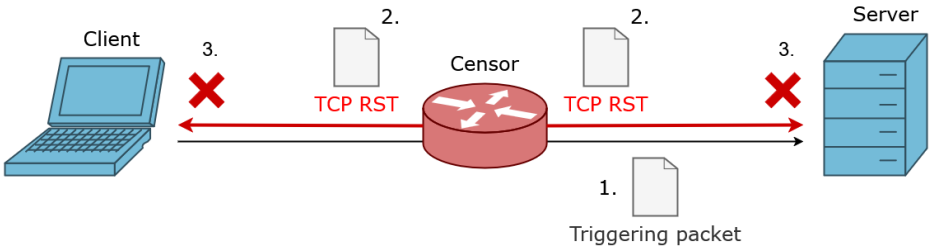

Figure 2.2: A TCP reset attack. The censor injects RST packets into the TCP connection in order to severe it and block all further communication between the client and the server.

TCP reset attack

A TCP reset attack can be used to severe a TCP connection between two endpoints. It is a common way for on-path censors in practice to block access to a resource after they have detected through other means, such as through an HTTP request, that someone is accessing unwanted content. A TCP reset attack can be thought of as a tool the censor can use to enforce DNS, HTTP, or HTTPS censorship.

A TCP reset attack utilises a normal and intended mechanism of the TCP protocol, namely the RST flag which is used to indicate the sending side has encountered some kind of error and it aborts the connection. Figure 2.2 depicts the steps in a TCP reset attack: First, a censor inspects a packet carrying a TCP segment passing through it in the network and uses some criteria to determine this is a connection it wants to censor. What criteria is being used depends on the application layer data being carried in the TCP segment. Second, the censor crafts two packets with TCP segments where the TCP RST flag is set, and sends the crafted packets to the client and the server or to only one of them depending on the implementation of the censor. The packets carrying these segments are referred to as **RST packets**. If the RST packets are correctly crafted, once the client or the server receives such a packet, it aborts the TCP connection without any further exchanges. The censor achieves its goal of severing the connection between the client and the server.

In order for a censor to accurately inject RST packets into the TCP connection, the censor needs to know the status of the connection [84] [17]. The censor needs to correctly utilize state information, such as sequence and acknowledgment numbers and what parts of the handshake have been completed, to craft the segments so that they look acceptable in the context of that particular TCP connection. If the crafted segments do not follow the protocol properly, they might be ignored by the end points and the connection does not

terminate. In practice this means censors have to maintain Transmission Control Blocks (TCB) that contain state information per connection. TCBs are an important part of the TCP protocol in general and not something specific to Internet censorship. The client and the server, also create their own TCBs as part of the protocol and update those as they send and receive TCP segments. A censor is said to be stateful if it maintains TCP state information.

DNS censorship

Before we introduce DNS censorship, let us recapitulate in slightly more detail how the DNS protocol works. DNS is an Internet wide distributed database that is used to map domain names to IP addresses [34] [35]. An example of a domain name is www.helsinki.fi where www. is the subdomain and .fi is the top-level domain. DNS is most often used when a user types a URL containing a domain name into their browser's search bar or clicks a link, and the browser sends out a DNS query to find out the IP address matching the domain name. The browser sends the query to a recursive DNS solver also known as a DNS resolver. The resolver is a DNS server that solves the client's query by contacting other DNS servers in the DNS resolution chain until the authoritative name server for the domain in question tells the recursive resolver the IP address of the domain name it requested. The resolver returns this answer to the client, and in case of web browsing, the browser now knows where to send the HTTP or HTTPS request to. DNS records can be cached in the browser, the resolver or elsewhere in the DNS chain to speed up the queries.

DNS was originally designed to be an unencrypted protocol and it most often runs on top of UDP for efficiency [34] [35]. DNS can also be used on top of TCP which is used for example when the reply is larger than the maximum UDP frame. The standard port for DNS is 53.

DNS queries can be censored primarily in two ways: either a censor is on the path between the client and the resolver, or the resolver itself does not return legitimate replies to the queries it receives. These two situations can also occur at the same time.

The case where the resolver itself is acting as the censor is depicted on the left side of Figure 2.3. When the resolver itself is the censor, DNS censorship works as follows: The user tries to access a website through a browser, and the browser sends a DNS query to a recursive resolver, such as the resolver of the user's ISP or an open resolver that answers queries made by any client. The resolver reads the domain name from the DNS query and

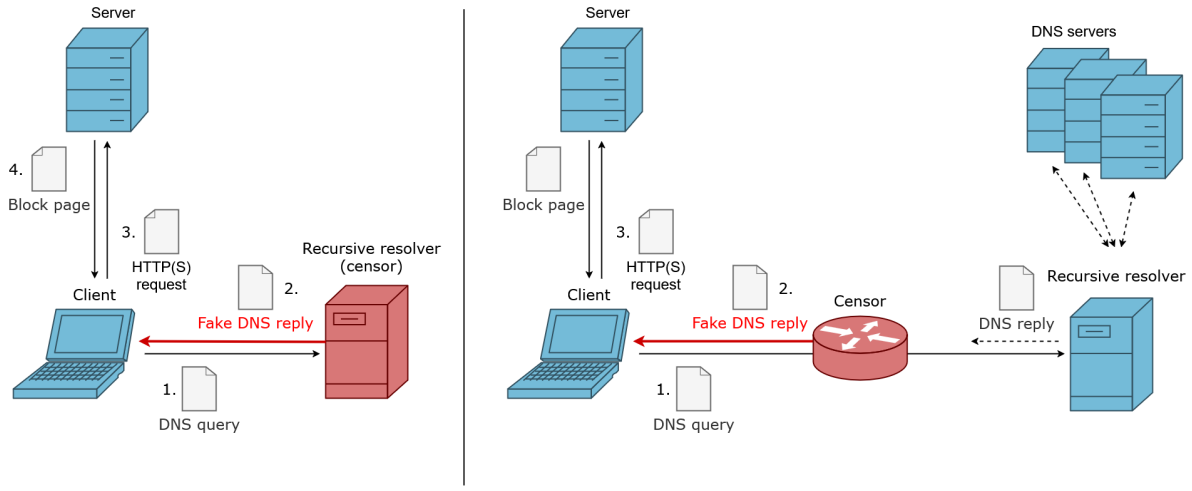

Figure 2.3: DNS censorship. The left side depicts a scenario where the resolver itself is the censor. The right side depicts a scenario where the censor is on the path between the client and the resolver.

checks this against a list of blocked domains or keyword rules it has [61] [47] [66]. The list can contain rules such as *.youtube.com, which would block youtube.com regardless of the subdomain used. If the domain name in the request matches a rule or a blocked domain on the list, instead of replying to the DNS query with the actual IP address of the requested domain, the resolver returns a false reply with an IP address that redirects the user to another server under control of the censor. This false reply will also poison the local DNS cache of the client. The server returns a blockpage that tells the user the site they were trying to access is being censored. Instead of a blockpage censors can also return DNS related error messages indicating the domain is non-existent or that the DNS server encountered an error, or return replies with private IP addresses [9] [66].

A user can choose the DNS resolver they want to use and switch to a resolver that does not censor DNS queries. However, it is still possible for the censor to be on the path between the client and the resolver and interfere with the queries. This is the overall model after which the GFW works. This type of DNS censorship is depicted on the right side of Figure 2.3. In this case, DNS censorship works in the following way [47] [8]: The user tries to access a website and the browser sends a DNS request as described above. The request is forwarded from router to router on its way to the DNS resolver, and at some point the request passes through the censor. Since the request is unencrypted, the censor is able to read the domain name written on the request. The censor runs a similar comparison to its list of blocked domains as described above and if the domain matches

a blocking rule, the censor moves to interfere. The censor crafts a fake DNS reply that contains an IP address that will redirect the user to a server that returns a block page. The censor sends this fake reply back to the client. In the most common case, the censor is an on-path censor and is unable to prevent the original DNS query from reaching the DNS resolver, which means the DNS resolver will also reply to the request. However, in the network sense the censor is usually located closer to the client and hence the fake reply will arrive earlier than the legitimate reply from the resolver. The client processes the first reply that arrives which is almost always the fake one that redirects the user to a blockpage. The client ignores the legitimate reply that arrives later.

If the client is using DNS over TCP, the censor has no need to inject a fake reply, but can use a more simple method to interfere with the DNS queries. After the censor perceives a DNS query for a blocked domain passing through it over a TCP connection, the censor can use a TCP reset attack to severe the TCP connection between the client and the DNS resolver before the client receives the DNS reply [54] [47].

HTTP and HTTPS censorship

In a similar manner, before introducing HTTP and HTTPS censorship, let us briefly recapitulate HTTP and HTTPS from the point of view of Internet censorship. HTTP is a well-known application layer protocol used most often to request webpages and other data over the Internet [39]. HTTP works in a stateless request-response manner where a client sends a request and the server responds. An HTTP request comprises of three main sections: a request line, possibly multiple headers, and an optional body section. From the point of view of Internet censorship, the request line, which contains the path to the resource the user requested, and the Host-header, which contains the domain name of the webpage, are the parts censors are interested in. Most often HTTP content is exchanged over TCP. Pure HTTP messages are exchanged in plaintext.

HTTP censorship works very similarly to how DNS queries are censored by an on-path censor between the client and the server [70] [54] [46]: A user attempts to navigate to a webpage on their browser either through a link or by typing the URL of the site to the browser's search bar. After the browser has resolved the domain name to an IP address through DNS and formed a TCP connection with the server hosting the webpage, the browser sends an HTTP GET request to the server requesting the webpage. The request travels through the network until it passes through a censor which maintains a list of

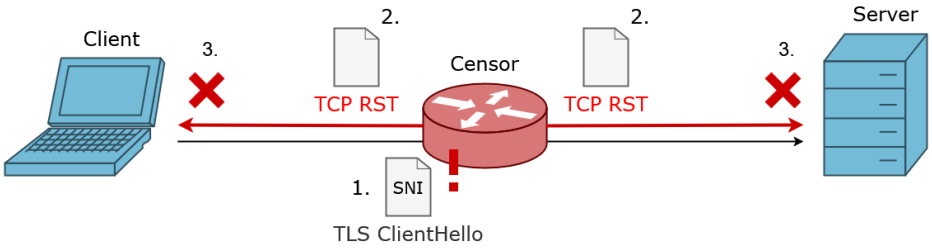

Figure 2.4: HTTPS censorship. The censor observers an unwanted domain name in the SNI header and uses a TCP reset attack to severe the connection between the client and the server.

blocked domains or keywords. The censor looks at the Host-header in the HTTP request, and if the Host-header matches an entry in the blocking list, the censor injects an HTTP response that redirects the user to fetch a blockpage created by the censor. The legitimate HTTP response that later arrives from the original server is ignored since the browser already received a response to the request. Instead of returning a blockpage the censor can alternatively sever the underlying TCP connection through a TCP reset attack. The censor can also consider other fields in the HTTP request in addition to the Host-header.

HTTPS is the secure version of HTTP [71], where the application layer data is encrypted using Transport Layer Security (TLS) [72]. This means that the TCP payload that contains the HTTP data is not transported in plaintext. In order to encrypt the application data, a handshake is needed. The TLS handshake is performed right after the three-way TCP handshake has completed. During the handshake, the client and server among other things authenticate the server and generate symmetrical session keys that are used to encrypt the content of the following communication.

From an Internet censorship point of view there is one extension header often included in the first TLS handshake message the client sends, the ClientHello, that is especially important. Many web servers host multiple domains which means websites can share the same IP address but have different domain names. Each domain has its own certificate so the server needs to know the domain the client is trying to access in order to present the correct certificate during the handshake. This domain name is given in an extension header of the ClientHello message called the Server Name Indication (SNI) [15]. The SNI header is sent in plaintext since not enough information has yet been exchanged for the handshake to be encrypted. This enables censors to read the domain name the client is trying to access even when the following HTTP data is encrypted.

Censor Method	% During Study Period	% All-Time
HTTP censorship	49	69
HTTPS censorship	41	44
Internet shutdowns	29	40
DNS censorship	24	46
IP or port blocking	9	30
Bandwidth throttling	6	13
Protocol fingerprinting	6	13
BGP attacks/disruption	1	11

Table 2.1: Internet censorship methods as measured by Master and Garman [61]. The numbers tell the percentage of studied countries that used the method during the study period as well as the percentage of countries that had used each method overall. Table modified from Master and Garman. In the original DNS censorship is listed as DNS tampering, HTTP censorship as HTTP/URL/keyword filtering, and HTTPS censorship as TLS-based filtering and the order of the rows is different.

HTTPS censorship works similarly to how HTTP censorship works; only the location of the requested domain name is different. The basic setup in HTTPS censorship is depicted in Figure 2.4. A user accesses an HTTPS website, and after a DNS resolution the browser forms a TCP connection with the server and starts the TLS handshake. A censor on the path between the client and the server inspects the packets passing through it, and if it perceives a ClientHello message with a forbidden domain in the SNI extension header, it uses a TCP reset attack to severe the TCP connection between the client and the server before the TLS handshake finishes and actual data is exchanged [20] [54] [46]. In most research, HTTPS censorship, TLS censorship, and SNI censorship, all refer to the same practice of censoring HTTPS connections through the SNI header.

Role of DNS, HTTP and HTTPS censorship globally

In 2023 Master and Garman [61] conducted a systematic literature review to gather data on which censorship methods were being used now or had been used in the past in different countries around the world. They studied around 70 countries most of which are high profile censoring countries and combined results from their literature review with quantitative data from different censorship measurement platforms.

Table 2.1 summarizes their results. According to Master and Garman HTTP and HTTPS censorship are the most prominent forms of Internet censorship globally [61]. Out of these

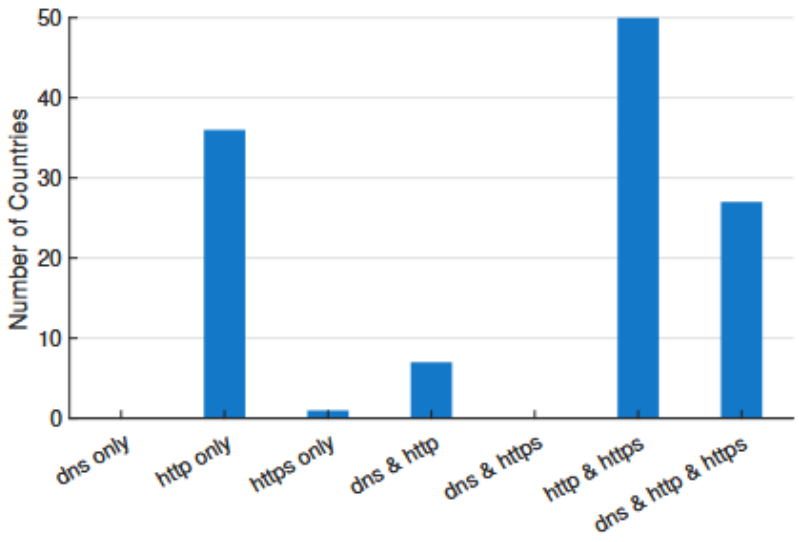

Figure 2.5: Popularity of DNS, HTTP and HTTPS censorship methods as measured by Jin et al. [54]. Figure by Jin et al.

two methods HTTP censorship is measured as the most used censorship method in the world. As can be seen from the table, HTTP censorship is not only the most used method during the study period but is also historically the most popular method. Popular in this context means that a method is being used in more countries than any other method. Due to the global scale of the study, the results only record whether a method is being used at all in each country, and it does not take into account for example how prevalent a method is compared to other Internet censorship methods within the same country.

These results are further supported by another global study by Jin et al. [54]. Compared to the study by Master and Garman, Jin et al. studied a much larger set of 177 countries and instead of a literature review, they gathered measurement data themselves through vantage points in each country. Their results are summarized in Figure 2.5. According to Jin et al. the most common combination of methods is using HTTP and HTTPS censorship together, although a sizeable number of countries utilize all three methods, DNS, HTTP, and HTTPS censorship simultaneously. Similarly to Master and Garman, their study also only recognizes whether a method is overall being used in a country.

According to both studies, while most high censoring countries censor both HTTP and HTTPS, in countries where less censorship or fewer methods are observed, HTTP is over-represented as the sole censorship method [61] [54]. This indicates that there is a number of countries that have not updated their Internet censorship systems to include HTTPS

censorship despite the prominence of HTTPS in modern web browsing. To better illustrate the role of HTTPS in the modern web, according to Google Chrome's own statistics [50], from the ten countries they used as representatives in their measurements, 88 - 99% of web traffic in early 2025 was HTTPS. Every country that wants to censor web traffic in any meaningful way needs to block both protocols, since only censoring one would provide an easy way to host censored content on websites using the other protocol. A positive interpretation of these results would be that multiple countries are unwilling to further invest in Internet censorship, even if they have done so in the past.

Master and Garman also found evidence of ongoing persistent DNS censorship in multiple countries, such as China, Iran, Myanmar and United Arab Emirates. However, according to their results, DNS censorship is not widely adopted around the world compared to other censorship methods, and no country they studied is using DNS censorship as its only Internet censorship method. On the global ranking of Internet censorship methods by popularity DNS censorship ranked fourth behind Internet shutdowns, HTTP-, and HTTPS censorship (Table 2.1). Jin et al. also arrived at the same conclusion: DNS censorship is not prevalent around the world. According to their results only China and Iran are in any meaningful way censoring regular DNS over UDP.

According to historical data Master and Garman collected, the role of DNS censorship seems to have diminished over time, which can also be seen from Table 2.1 when comparing the All-Time column with the percentages observed during the study period [61]. The authors speculate that this might be due to there being multiple relatively easy and effective methods to circumvent DNS censorship, which might make DNS censorship seem unworthwhile for most censors to implement and manage in today's Internet. It might also be that HTTP(S) based methods are deemed sufficient and more proficient at the task.

Other Internet censorship methods

For the sake of completeness, we should mention a few other prominent Internet censorship methods, that will not be covered in detail in this thesis but still deserve a mention. Table 2.1 provides a good overlook of the diverse set of Internet censorship methods that governments around the world can utilise. In addition to DNS, HTTP and HTTPS censorship, other Internet censorship methods include Internet shutdowns, IP address blocking (listed in Table 2.1 as IP or port blocking), and bandwidth throttling. Protocol

fingerprinting is an advanced method also employed in the Great Firewall of China, and we cover protocol fingerprinting in Chapter 5. Border Gateway Protocol (BGP) attacks are not very common and we will not explain them here.

Government ordered Internet shutdowns have become an increasingly common method to censor the Internet [13] [55]. An Internet shutdown is an intentionally caused outage where access to Internet is entirely removed from an area for a controlled amount of time typically during political unrest with the goal of making digital communication and information sharing difficult. Despite Internet shutdowns being a prominent censorship method, shutdowns are not very common in China [25].

IP address blocking has historically been an important method to censor the Internet [54] [61]. IP address blocking refers to a censor maintaining a blacklist of IP addresses that it blocks access to. If a censor notices a packet with a blacklisted IP destination address passing through it, it moves to interfere with the connection. IP address blocking was more widely used in the past, but on the modern Internet it can easily cause too much collateral damage if used carelessly. A single web server with one IP address can host multiple web domains, and if the censor blocks the server by its IP address with the intention of censoring one unwanted domain hosted on the server, the other domains also become blocked as collateral damage. Despite this, some censors, including the Great Firewall of China, still utilize their existing IP address blocklists [23], and IP address blocking can also be used as a complementary method to make more complex censorship methods efficient through something called **residual censorship** [91].

Residual censorship is a functionality that well-resourced governmental censors such as the GFW can employ: after detecting and blocking an unwanted connection through protocol specific means, such as HTTPS censorship, the censor records the triple (client IP address, server IP address, server port) and blocks all further communication associated with that triple with TCP RST packets to both directions for a short period of time [16]. Residual censorship is likely a mechanism that helps to save computing resources of the censor: the censor can replace a computationally more expensive analysis with a simple lookup for the consequent requests between the same client and server that are likely to follow.

Port blocking refers to blocking all traffic towards a certain port [61]. Port blocking enables censors to essentially block whole protocols, since blocking all traffic towards the HTTPS port 443 for example would censor all traffic using that port.

Bandwidth throttling is the act of artificially considerably slowing down traffic towards or originating from selected targets on the Internet, so much so that the target becomes

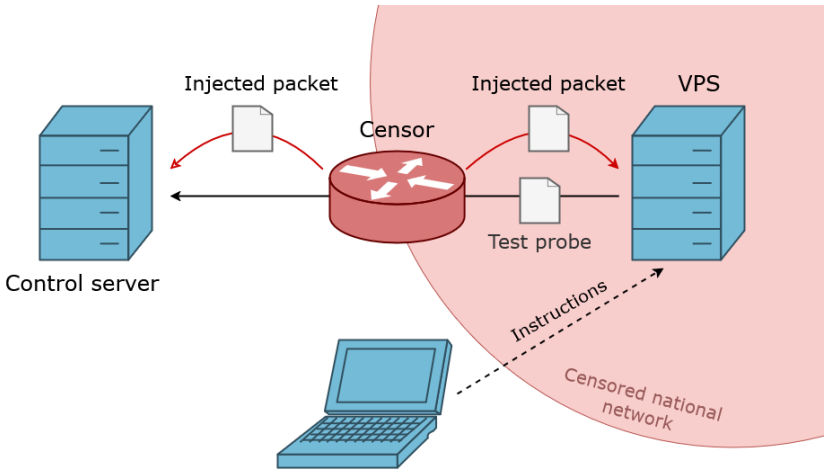

Figure 2.6: Probing a packet filtering system from inside the censoring country. The observer connects to a VPS inside the country and sends probes to a server outside of the censoring country to test for censorship. Blue nodes are under control of the observer.

practically unusable from the censored region for a period of time [61] [92]. A typical target would be a social media platform. The benefit of bandwidth throttling is that it is a relatively invisible censorship method, as from the users point of view it might seem like the issue is with the platform itself.

2.3 Studying packet filtering systems

Most countries do not publish information about their Internet censorship systems or how they work. This means censorship systems such as the GFW are essentially a black box to anyone else except their operators, and research is often the only way to infer information about how these systems work. In this section we describe a common research setup used to study packet filtering systems. This high-level description is based on the research articles used throughout the thesis, and we cite a non-exhaustive list of prominent examples here [47, 9, 53, 70, 20, 46, 23, 37, 36, 6, 91, 54]. In this discussion we call the entity researching the censorship system an observer and refer to the censoring regime simply as the country.

Studying the structure and behaviour of a packet filtering system usually involves sending packets through the censoring devices and measuring the response against the expected result. If the component of the censorship system under test, such as HTTPS censors, work bi-directionally, the observer can try to probe the system both from inside and outside to gain the most information.

Probing the system from inside the country commonly requires a few nodes. The setup described here is depicted in Figure 2.6. The first node is a computer outside of the country that the observer can use to connect to a Virtual Private Server (VPS) or another computer under control of the observer located inside the country. The observer uses this computer to send test instructions to the VPS. The VPS is then used to send test probes through the censorship system to simulate how a citizen of the country would request a resource hosted outside the censoring country. The test probes pass through the packet filtering system, aiming to reach either a public web server hosting the requested resource or a control server of the observer. The test probes may or may not solicit a response from the packet filtering system while passing through it. The result, either the legitimate target resource or an injected response by the censors, is captured by the VPS and possibly by the control server. The received reply is recorded and compared to the expected result.

Probing the system from outside of the country can be done with a similar setup, but the probes are naturally sent from the observer's computer outside of the country to a VPS or a public web server inside the censoring regime. The results can then be collected from the probing computer and possibly also from the VPS inside the country.

Internet censorship research commonly studies either what or how something is being censored. When the focus of a study is on what is being censored, it means aiming to accurately find out the blocking rules or domain lists being used in the censoring nodes. Often the only way to find out what domains are being blocked is to send a large number of queries for different domains through the censoring system and see which ones are being censored and which ones are not. Identifying the expected result without censorship is not always simple due to for example geotargeting. Since it is practically impossible to test all domains against a system, it is necessary to form some form of base pool of domains.

A common way to form these base pools is to use some global domain ranking that ranks domains based on their popularity. Before being discontinued in 2022, a common choice of domain ranking for research purposes was Amazon's Alexa Traffic Rank. The practice of using global domain rankings in research has also been questioned, due to how different rankings rarely agree on popularity of domains and how there has been evidence of how the rankings are susceptible to manipulation [68]. To counter this, Le Pochat et al. developed the Tranco ranking [2] that aims to be a more research oriented domain ranking that is hardened against manipulation and that combines ranking results from multiple global domain ranks to achieve more accurate results. After its creation the Tranco ranking has been used in multiple Internet censorship studies.

In addition to testing popular domains, for country specific studies, research groups often include country specific domain lists into the base pool, such as local lists provided by Citizen Lab [27]. These types of regional domain lists typically comprise of domains that are known to contain material on sensitive topics that have been reportedly been censored in the country in question.

After probing a system against a base pool, it is possible to gather statistics such as the percentage of censored domains and to classify the censored domains to gain more insight on how Internet censorship is being used in that particular country. A domain categorization tool from FortiNet called FortiGuard [85] is commonly used in Internet censorship research to categorise websites or individual pages based on their "dominant content". The FortiGuard categorisation includes categories such as "news and media" and "personal websites and blogs" for example. How the base pool was formed and at what time influences results of domain based analysis and is something that needs to be taken into account especially when comparing results between studies made by different groups.

2.4 Internet censorship landscape in China

China has been recorded to be one of the most aggressive censors in the world [54] [61]. In the 2024 Freedom on the Net ranking by an American non-governmental organisation Freedom House the six worst offenders of global Internet freedom were China tied with Myanmar, then Iran, Russia, Cuba, and Vietnam, with China being the worst offender for the 10^{th} consecutive year [41]. As a result, China has for years been a major target of Internet censorship research.

The Internet censorship landscape in China is characterised by the thorough governmental control over Internet infrastructure, national companies and information on citizens [95]. Internet censorship in China is so efficient precisely due to how censorship is enforced both through technical means in the Internet backbone through the Great Firewall and in social media applications, as well as in legislation that binds both individual citizens and businesses. Countries such as China and Russia follow an ideology that has been called "cyber sovereignty", according to which countries should be allowed to control their national networks similarly to how they control their physical borders [75].

Multiple government ministries together oversee the enforcing of digital censorship in China and the management of the Great Firewall also falls under their responsibility [25]. Governmental control through ministries ensures censorship policies are uniformly

implemented throughout China.

There are only few internet exchange points that are all controlled by the government, and all ISPs connect to the outside world through these [25]. The largest ISPs, China Telecom, China Unicom and China Mobile, are state-owned and dominate the telecommunication market [46] [25] [24]. ISPs, cloud providers and other companies in China are legally obliged to hand over their data to the government for inspection if asked [95] [57]. Internet companies periodically receive orders from the government to censor certain topics and a company faces considerable negative consequences in case it fails to comply [25]. VPNs are heavily regulated and can be used only after government approval for specific purposes. It is illegal to use an unapproved VPN, which means many Chinese and international commercial VPNs are illegal in China. It is also illegal to spread information about or provide services to bypass the GFW.

Building of the Great Firewall apparently started in the late 90's under the name of "the Golden Shield Project" [47]. Internet censorship has been built into the Internet infrastructure of China since the beginning [95], and this is likely one of the reasons why China has managed to build such sophisticated and comprehensive censorship systems.

Even though the intentions of the Chinese government are not clearly documented, the purpose of the GFW can be inferred from the sites and topics it censors: The purpose of the GFW is to block Chinese citizen's access to selected resources abroad with the goal of controlling information about the government [70] [95] [25]. Another purpose seems to be to make civil organizing and protesting more difficult [56] [95]. The role of the GFW is to block popular foreign applications and services such as Facebook, Instagram, Twitter, and YouTube, since the Chinese government cannot easily control the content posted on these platforms, which forces most Chinese people to use domestic alternatives for social media that are strictly monitored and censored through social media censorship [26].

Some topics have been controversial in China for a very long time and tend to always be censored. Permanently taboo topics in China include the Tiananmen square protest and massacre, independence of areas under the Chinese rule such as Taiwan and Tibet, democracy and multi-party systems, the Falun Gong movement, human rights, dissidents, civil organizing and censorship circumvention [70] [95] [28]. Foreign news sites that cover Chinese issues are often censored. More recent topics the Chinese government wants to strictly control the narrative of include COVID-19 and the oppression of the Uighur ethnic minority in Xinjiang [70]. In addition to these topics for example porn is actively censored in China [42] [43]. The degree of digital censorship is not uniform across China

as for example the Uighur minority in Xinjiang is censored and monitored much more excessively and invasively than majority Chinese citizens [29].

In authoritarian countries the amount of Internet censorship and tactics used to employ it often fluctuate. Many countries increase the amount of censorship or implement harsher policies during politically sensitive times, such as during elections, protests or mobilizations [13] [32], or as a reaction to controversial events [57] [10]. China is not an exception and has been recorded to tighten its censorship policies during political events such as important meetings of the Chinese Communist Party (CCP) and conversely to censor less aggressively after such events are over [6] [91].

Internet censorship by the GFW seems to be limited to mainland China: areas under the influence of the Chinese government, such as Hong Kong and Taiwan, are almost never affected by the GFW and its censorship [70].

Social media censorship is prominent in China and it is an area of digital censorship that has been well researched [10] [56] [93] [57]. Even though the focus of this thesis is on the GFW, for the sake of completeness, we discuss social media censorship here briefly and will not come back to this later in the thesis. While social media censorship is different from Internet packet filtering, they are complementary tactics used to achieve control over information in every corner of the Internet: the GFW censors content that is hosted outside of China, while social media censorship is used to censor content hosted on domestic Chinese applications and services. It can be argued that the GFW is what makes social media censorship in China so effective, since it forces normal Chinese citizens to use censored domestic applications instead of global alternatives.

Social media censorship targets posts, videos and audio by looking for banned keywords or other signs of forbidden content [10] [56] [93] [57]. Content or accounts that are flagged for containing sensitive keywords are hidden or removed. Social media companies in China have to comply with national censorship laws that make them legally responsible for content posted on their services [95] [57]. These laws are often purposefully vague and can be used arbitrarily, pushing companies to over-censor, since it is safer. Many applications and public computer services such as Internet cafes implement real ID identification to easily trace any activity back to a real person [29]. In addition to social media sites, search engine results [10] and email are also censored [58]. Search engines that do not comply with the national censorship laws cannot operate in China. As an example, Google used to be available in China, until it was driven out of the country by the government due to Google allowing its customers access to an uncensored version of their search engine [24].

3 DNS censorship in the Great Firewall

In this chapter we take a look at DNS censorship in the Great Firewall of China and what methods have been studied to circumvent it. The Great Firewall is a packet filtering system that affects traffic travelling between China and the rest of the world, which means a Chinese user would encounter DNS censorship of the GFW when trying to use an open DNS resolver outside of China. In Section 3.1 we discuss how DNS censorship has been implemented in the GFW, what kind of domains are censored through DNS censorship, and how DNS censorship of the GFW has been studied. In Section 3.2 we take a look at what research has been conducted about DNS censorship carried out by DNS resolvers themselves, since this affects the comprehensiveness of DNS censorship and available circumvention options in China. In Section 3.3 we discuss what DNS specific ways have been developed to circumvent DNS censorship, the most prominent of such methods being encrypted DNS.

3.1 DNS censorship

Records of the GFW using DNS censorship can be traced back to at least year 2002 [8]. At the time DNS censorship provided a more permanent and flexible way to block a website as opposed to simply statically blacklisting the website's IP address [60]. According to early studies focusing on DNS censorship in China by Lowe et al. in 2007 [60] and Anonymous et al. in 2014 [8], it seems the DNS censorship in the GFW stabilized to its current model quite early on before year 2007. The system has not undergone major architectural changes since then.

China is globally the number one country to censor through DNS, both over UDP and TCP [54]. This is likely one of the reasons why DNS censorship of the GFW has been the target of much research despite how DNS censorship is globally a relative unused method: China is one of the few countries in the world that uses DNS-based techniques to enforce large-scale national censorship [61].

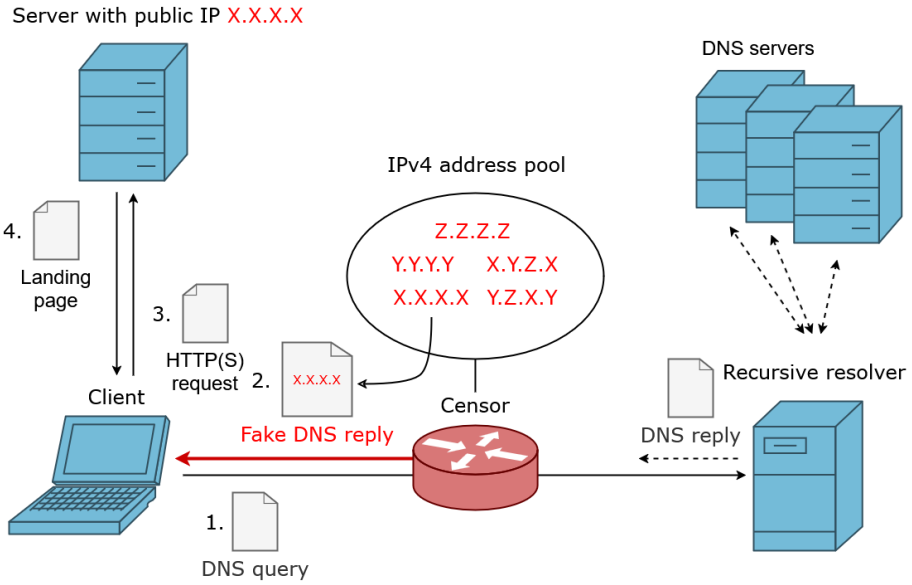

Figure 3.1: DNS censorship by the GFW (IPv4). Instead of directing the user to fetch a blockpage, DNS censors of the GFW return publicly routable IP addresses that do not belong to it. This directs to user to unexpected or non-existent landing pages.

DNS censorship model

The GFW's DNS censorship system is an on-path censor that censors DNS over UDP by injecting fake replies to DNS queries that contain blocked domain names or keywords [47] [9] [8]. When censoring DNS over TCP the GFW uses the most common method in general to block any communication over TCP, a TCP reset attack [47] [54]. The system is bi-directional which means it censors queries passing through it in both directions, and the system can be probed from both inside and outside of China. The GFW seems to only block DNS queries sent to the standard port 53 and sending DNS queries to other ports does not trigger the censorship system [47] [9]. The DNS censors seem to be located at border ASes in China [8].

Even though the way the GFW censors DNS queries over UDP follows the common model described in Section 2.2, the way the GFW crafts its fake replies is unusual. Most other countries with DNS censorship systems return an explicit block page that explicitly tells the user the domain they were trying to access is blocked [9]. This gives a sense of transparency. However, to IPv4 queries (type A queries), the GFW returns publicly routable IP addresses usually belonging to large American companies such as Facebook,

Twitter or Dropbox [47] [9] [54]. The DNS censorship model of the GFW is depicted in Figure 3.1. The addresses are returned from a periodically changing pool that consist of around 1800 IPv4 addresses [47]. Around 600 of these addresses make up 99% of the IP addresses used in the fake replies. There seems to be a connection between the requested domain and the IP address returned in the fake reply, since querying specific domains elicits responses from specific subpools of IP addresses instead of the whole pool [9] [47]. This practice of returning publicly routable IP addresses makes it more challenging from Internet censorship research point of view to automatically detect when a reply is fake, since the landing pages can contain varying content instead of a static blockpage.

For IPv6 queries (type AAAA queries) the GFW simply returns a non-existent IP address [9] [47]. For one DNS query it is possible to receive a fake reply from multiple DNS censors which increases the chances of a fake reply being the first to make it to the client and also helps cover for cases where a single censoring device is unable to handle all requests passing through it.

The DNS censorship system manages to block the censored domains practically every time when the query is sent from inside China [47]. It seems there are sometimes cases when the legitimate reply arrives first and this is suspected to be due to the censors having reached their computational limits, but according to Hoang et al. [47], this happens as rarely as 1% of the time. The system seems to be slightly less effective in censoring queries that come from outside of China to DNS servers inside China in which case the fake reply arrives first 89% of the time.

The blocking rules of the DNS censorship system are revised and updated from time to time [47]. It seems likely the system is managed both automatically and manually, since most of the time domains are blocked very quickly after they have been created on the Internet, but in some cases it can take multiple months for a domain to be censored.

Censored domains

There are multiple quite comprehensive studies from recent years which have focused on the GFW from different perspectives. One area of key interest when studying packet filtering systems is the domains the system censors, and this is case with the GFW as well.

In 2021 Hoang et al. [47] developed a longitudinal measurement platform called the GFWatch that measures the GFW's DNS censorship by testing domains daily and report-

ing the domains' status as censored or uncensored. Hoang et al. built their base pool of domains by combining top-level domain zone files with Citizen Lab test lists [27], the Tranco ranking [2], and domains from the Common Crawl project [30] that provides free web crawl data to researchers. Their base pool is quite diverse and extensive, and due to including top-level domain zone files has the advantage of staying up-to-date as domains are created on the Internet. They report testing on average 411 million domains each day.

Hoang et al. probed the GFW from outside of China from an academic network in the USA towards two hosts under their control within China in different ASes [47]. The target hosts have no DNS resolution capabilities, which means that all responses are injected by the GFW. If any censorship is detected, they verify the censorship by querying the same domains in reverse direction from within China towards their control servers in the USA. A domain is tested three times a day to counter false negatives that might arise from packet loss or from the GFW occasionally failing to censor a blacklisted domain. They only test DNS over UDP.

According to their results the GFW censored more than 331 000 domains through DNS censorship in 2021 [47]. This is likely the largest number of censored domains discovered and tested in research focusing on DNS censorship of the GFW. Hoang et al. estimated that around 41 000 of the censored domains were overblocked and did not host any sensitive information, but were rather blocked as collateral damage due to the domains accidentally matching blocking rules.

Hoang et al. analysed the censored domains and categorised them with FortiGuard. 45,5% of the blocked domains were unpopular smaller websites that were FortiGuard could not categorise. Otherwise the largest categories of blocked domains were "business", which includes general business sites (examples of sites belonging to this category are linkedin.com or alibaba.com), "pornography", and "information technology", which includes sites whose main content is computer or mobile phone peripherals or services (examples are apple.com or stackoverflow.com). Business and information technology were the two most represented categories in Alexa's top 1 million domains at the time [9], so domains from those ategories were likely well represented in the Tranco ranking that Hoang et al. used as part of their base pool. Hoang et al. categorised domains only after reducing each domain to its shortest fully censored parent string in order to avoid popular sites with numerous subdomains dominating the results. As an example, instead of categorising censored domains en.wikipedia.org and zh.wikipedia.org separately, only their common parent string wikipedia.org is categorised.

	Hoang et al.	Anonymous et al.	Jin et al.
Tested domains	534 000 000	1 000 000	$1908 - 3953^{*}$
Censored domains UDP	311 000 (0,06%)	$23\,995 - 24\,636^{**}$ (2,4%)	366 - 759 (19,2%)
Censored domains TCP	-	-	384 - 795 (20,1%)
Top categories	1. Uncategorised 2. Business 3. Pornography	1. Proxy avoidance 2. Personal websites 3. Explicit violence	1. News and media 2. Search engines 3. Proxy avoidance

Table 3.1: DNS censored domain results from Hoang et al. [47], Anonymous et al. [9] and Jin et al. [54]. Number of censored domains are given for DNS over UDP and DNS over TCP if reported. Percentage of censored domains over all tested domains is given in parenthesis for each result. *The number of tested domains is not reported for China but is between these numbers. **The number of censored domains increased during the study period.

Anonymous et al. [9] studied the GFW's DNS censorship around the same time in 2020. They executed their tests by sending DNS queries from an academic network in the USA towards a VPS they controlled inside China. Anonymous et al. report the GFW to be censoring 23 995 domains, which had increased to 24 636 towards the end of their nine month study period. They also categorised their test domains with FortiGuard and they used Alexa's top 1 million domains from 2019 as their base pool. In their results, "proxy avoidance", which includes sites that provide information on how to bypass Internet censorship, "personal websites", which are websites hosted by individuals where the main content is the owner's own ideas, and "explicit violence" are the most censored categories of DNS blocked domains in China. These results clearly differ from Hoang et al.'s results. The high number of personal blog sites being censored in Anonymous et al.'s results is possibly due to the GFW's DNS blocking list containing subdomain rules such as *.thumblr.com that match a large number of personal blog sites combined with the fact that Alexa's domain rankings often contain sites from popular blog services [47]. Anonymous et al. presumably only tested DNS over UDP but they do not mention it explicitly.

A third study from recent years that reports on DNS censorship in China is the global study by Jin et al. [54] that was briefly introduced in Section 2.2. Jin et al. conducted DNS over UDP queries from RIPE Atlas probes [74] and DNS over TCP queries from SOCKS proxies within China, towards their control server outside of China. Both Atlas probes and SOCKS proxies are essentially networks of nodes that can forward requests.

Jin et al. formed their base pool of domains for each country by combining Alexa's top 1000 domains with the global sensitive domains list and country specific lists from the Citizen Lab. They do not report on the number of censored domains in China, only that their base pool size varied by country (between 1908 and 3953 domains) and that China censored 19.2% of tested domains with DNS over UDP and 20.1% with DNS over TCP. They used FortiGuard for domain classification and found "news and media", "search engines and portals" and "proxy avoidance" to be the three most censored categories in China. Since Jin et al. measured DNS censorship over both UDP and TCP, they report that blocked domains in China were mostly consistent between the two protocols.

The relevant results from the three above studies are summarised in Table 3.1. The results regarding the number and categories of blocked domains are not consistent between the three above studies, even though all studies were conducted around the same time and used similar probing setups. The difference in number of censored domains is naturally dependent on the size of the base pool. Other than that, which domains were selected to form the base pool seems to greatly affect the categorisation results. This highlights how domain related Internet censorship results at least with currently prominent test methods are not easily comparable between studies and cannot be used to derive definite conclusions. Since Hoang et al. tested the highest number of domains from diverse sources it is safe to assume the GFW censors at least 311 000 domains and that this number includes domains the other studies did not discover.

Interestingly, instead of only censoring the Internet for Chinese users the GFW has also been recorded to geoblock foreigners from accessing Chinese websites [47]. Hoang et al. report that some of their test queries targeting Chinese websites that originate from outside of China solicit a fake reply from the GFW, but can be normally accessed from inside of China. The website they used as an example of geoblocking was a governmental website. Currently there are no records of any large scale geoblocking, but it is certainly something the DNS censorship system can be used for if the Chinese government wishes to do so.

3.2 DNS censorship by resolvers

Let us briefly take a look at research that has focused on censorship carried out by the DNS resolvers themselves, globally or in China. This type of censorship is not directly carried out by the GFW but since it affects the comprehensiveness of DNS censorship and available circumvention options in China, it is still a topic of interest to people studying

DNS censorship in China.

Pearce et al. [66] studied open DNS resolvers in 2017 from an Internet censorship point of view and measured that globally only 0,31% of DNS responses they received from open resolvers were manipulated. 88% of tested DNS resolvers did not manipulate any results. A response counted as manipulated if it failed to meet their consistency standards, where the idea was to take into account that a legitimate reply does not necessarily always contain the same IP address but the address in the reply can come from a pool of addresses owned by that domain. For testing DNS resolvers, they developed a system called Iris that sends queries to a large set of open DNS resolvers around the world. Their domain base pool was combined from the Citizen Labs lists of sensitive domains, presumably the global one, and approximately the same number of domains randomly selected from Alexa's top 10 000 domains. It is unclear where geographically the system was deployed during their tests, but it seems very unlikely they ran their experiments from within China.

Globally, most of the manipulated results they received were from Iranian resolvers, with China being second [66]. The aggregated median for manipulated responses per a Chinese resolver was around 5%. Responses from Chinese resolvers accounted for 1% of all responses, but 15% of all manipulated ones. However, it seems that their tests cannot differentiate between local manipulation by the resolver itself and country-wide manipulation such as injections from the GFW. Thus it seems likely their results for Iran and China actually represent DNS injections by national firewalls and not by open resolvers in the country. This is further supported by the fact that the manipulated results they receive from China characteristically contain a public IPv4 address more than 99% of the time. On the other hand, the domains they query from Chinese resolvers are censored quite inconsistently, some only 50% of the time, which seems unlike previously recorded behaviour of the GFW.

In order to draw definite conclusions on whether open resolvers within China manipulate their responses, the DNS queries would have to originate from China and not transit through border ASes. Due to the legislation in the country, it can be expected that both resolvers owned by ISPs and open resolvers have to censor their results or they cannot operate.

Results from Niaki et al. [62] further support the conclusion that open resolvers around the world largely do not censor their results. Niaki et al. run a longitudinal Internet censorship measurement platform called ICLab that measures different forms of Internet censorship from vantage points around the world. They reported in 2020 that while DNS resolvers in

at least 56 countries, both open and local (presumably ISP owned), do manipulate their results, they do so in very small numbers. They do not give specific results for China.

3.3 Circumventing DNS censorship: Encrypted DNS

If a Chinese user is subjected to DNS censorship when using a domestic DNS resolver (such as depicted on the left side of Figure 2.3), the censorship can be evaded by changing resolvers to a non-censoring open resolver outside of China. However, this causes the DNS query to pass through the GFW which censors the query either by injecting a fake reply (see Figure 3.1) or by severing the client's TCP connection to the resolver. This means the censor on the path between the client and the resolver needs to be defeated in order to successfully circumvent DNS censorship in China. Probably the most prominent DNS specific method that has been studied for this purpose globally is **encrypted DNS**: either DNS over TLS (DoT) [51] or DNS over HTTPS (DoH) [49].

In DoT the DNS messages are encrypted with TLS encryption so that the requested domain and other information is not sent in plaintext [51]. DoT also protects the integrity of the messages so they cannot be tampered while in transit. DoT works over UDP and operates on port 853. In DoH the DNS queries are wrapped in HTTPS requests, usually in the body of a HTTP POST request [49]. As was recapitulated in Section 2.2, HTTPS uses TLS to encrypt the HTTP payload, so DoH messages are also encrypted using TLS. DoH uses the standard HTTPS port 443. Both methods encrypt the communications between the client and the recursive resolver. If a DNS resolver is capable of either DoT or DoH it is called an encrypted DNS resolver. On the client side using DoT or DoH typically requires configuring a browser to use encrypted DNS.

From a censorship perspective, the key difference between DoT and DoH is that DoT uses a distinct port which makes it easy to identify when someone is using DoT. That does not affect the encryption, but it is possible for censors to block the usage of DoT completely by blocking all traffic on port 853. Trying to censor DoH by blocking the HTTPS port 443 would block all other HTTPS traffic as well, which is not a desirable state for censors. This possibly makes DoH the better encrypted DNS from Internet censorship circumvention point of view.

Jin et al. studied the effectiveness of DoT and DoH from Internet censorship point of view in 2021 [53], and one of the countries they studied was China. In order to find out whether DoT or DoH could help circumvent DNS censorship in China, they made encrypted DNS

queries for sites under DNS censorship and measured which sites became available by using encrypted resolvers outside of China. They sent the queries from four cloud servers in geographically different locations in China towards encrypted resolvers that are known to not manipulate results.

Overall, they were able to unblock 37% of the sites they tested with the help of encrypted DNS [53]. The domains that were unblocked by the usage of encrypted DNS were mostly categorized by FortiGuard as sites related to "proxy avoidance", "pornography" and "news and media". The remaining 63% are domains that are further blocked by other methods, mostly by the GFW's HTTPS censorship. A large number of sites are also directly blocked based on their IP addresses. Switching to a different encrypted resolver did not affect the results for a specific vantage point within China, which is to be expected.

Using encrypted DNS resolvers can help defeat a DNS censor on the path between the client and the resolver, but it achieves complete circumvention of DNS censorship only if the encrypted resolver itself is not manipulating the results. With this in mind, Jin et al. also studied censorship carried out by the encrypted resolvers themselves [53]. They studied 3813 DoT resolvers and 75 DoH resolvers and observed that globally around 1.5% of all encrypted DNS responses were manipulated. 2/3 of studied resolvers censored at least one domain, which is a considerably higher percentage compared to unencrypted resolvers [66]. Significant percentages of globally manipulated domains seemed to be related to blocking of adult content and gambling, or information technology sites. The authors speculate this to be the result of encrypted DNS resolvers commonly offering options to filter sites unsuitable for underage users. Overall, manipulation of DNS responses by encrypted resolvers does not seem to be politically motivated and encrypted resolvers can likely be used to assist in circumvention of governmental Internet censorship.

In order to hinder the privacy increasing effect of encrypted DNS resolvers, some open resolvers are directly IP address blocked by the GFW and cannot be accessed from within China. Basso [12] studied the availability of global encrypted DNS resolvers from within China in 2021 by testing 123 DoT or DoH resolvers. According to the results, 93% of DoT queries and 89% of DoH queries succeeded. Google's and Cloudflare's DNS encrypted resolvers were the most blocked, which is supported by results from Jin et al. who also report some of Google's encrypted DNS resolvers being blocked in China, for example the encrypted version of the famous 8.8.8.8 [53]. The results indicate that a significant number of encrypted resolvers are still accessible from within China and can be used for censorship circumvention.

Even though encrypted DNS can assist in circumvention of DNS censorship, encrypted DNS also has some technical weaknesses: Encrypted DNS only protects the messages exchanged between the client and the recursive resolver, so it is still technically possible for a censor to interfere with messages exchanged between the resolver and the other DNS servers, which could cause the resolver to cache and return a manipulated IP address [53]. This type of cache poisoning does not seem very practical for governmental censors around the world, but it is a technical possibility.

It is also possible to accurately fingerprint websites solely based on their observed destination IP addresses, which decreases the privacy increasing effect of encrypted DNS [48]. Hoang et al. reported in a 2021 study how they successfully fingerprinted 95% of sites they categorized as both popular and sensitive solely based on the sites' destination IP addresses. So far there is no evidence of such fingerprinting methods being used for censorship purposes.

4 Web traffic censorship in the Great Firewall

In this chapter we cover HTTP and HTTPS censorship in the Great Firewall and what protocol specific methods have been studied to circumvent either type of censorship. In Sections 4.1 and 4.2 we discuss how HTTP and HTTPS censorship respectively have been implemented in the Great Firewall. In Section 4.3 we cover an HTTPS specific circumvention method called ESNI. In Section 4.4 we discuss TCP-based circumvention techniques that have been used to circumvent HTTP and HTTPS censorship, as well as a similar but new TLS-based method.

HTTP censorship has historically been an important form of Internet censorship in China and early studies of HTTP censorship by the Great Firewall date back to 2002 [94] [95] [28]. As websites around the world started to transition from HTTP to HTTPS the GFW had to reflect this change. It seems the GFW started to censor HTTPS by the SNI header somewhere between 2017 and 2019 [84] [23]. Today, the focus of web traffic censorship is on HTTPS. According to Cloudflare's statistics [5], in February 2025 around 83% browser traffic from China towards websites hosted on Cloudflare's servers was HTTPS. According to Jin et al. [54] China is the second biggest HTTP and HTTPS censor in the world right behind Iran.

4.1 HTTP censorship

The GFW censors HTTP by looking for banned keywords in the HTTP requests, either on the request line or the Host-header [70]. After an HTTP censor has detected a banned keyword in the request, the censor uses a TCP reset attack to severe the connection. Much like in its DNS censorship, the GFW does not return a blockpage that would indicate to the user the site they are trying to access is being censored, but utilises TCP reset as a less transparent way of blocking the connection. The GFW censors HTTP on all TCP ports.

After the censor has injected RST packets, residual censorship is used to block all further communication between the server and the client for 90 seconds [70] [84]. It seems in the

early stages of HTTP censorship in the GFW, residual censorship could last for as long as 30 minutes and also block access to uncensored sites during the penalty period [94]. Such drastic censorship was likely deemed excessive as Internet usage became more widespread and part of everyday life.

Rambert et al. studied the GFW's HTTP censorship in 2021 [70]. They sent HTTP requests between multiple VPS:es both from outside and inside of China, and inserted sensitive keywords into different places in the requests. Rambert et al. used a rather unconventional method of choosing which keywords to test: they extracted sensitive keywords from titles of Wikipedia pages in English, Standard Chinese, and three other Chinese language variants, and combined these with keywords that were recorded by Xia Chu [31] to have been blocked by the GFW's HTTP censors in 2014. The keywords by Xia Chu were also selected from Wikipedia pages. On top of these two keyword lists they used a list from CitizenLab [26], that contains keywords that are censored in popular chat applications in China. Overall, their keyword list contains a total of 17 366 words all of which are expected to be censored in China.

Rambert et al. discovered by accident during their probing tests that the GFW seems to maintain two separate lists of censored keywords the HTTP requests are checked against [70]. The first of the lists is a static list that triggers censorship every time a request contains a word from this list. The second, much longer list, contains keywords that trigger censorship only if the request line contains the English word "search". This indicates that the HTTP censorship system has at least at some point been focused on blocking access to foreign search portals with the aim of making it difficult to search for information on forbidden topics. It is possible that different keyword lists are also used for inbound and outbound traffic, since traffic entering China seems to be more strictly censored compared to traffic leaving China.

Even though HTTP is less relevant in today's Internet, the HTTP censors' keyword lists are still being actively updated [70]. Rambert et al. repeated their probing tests weekly for a month and observed changes in the censors' keyword lists with a median number of 270 weekly additions and 400 removals per VPS acting as a HTTP client. This seems like a very high number of weekly changes to the keyword lists. They also retested the 451 keywords from Xia Chu that were all censored in 2014 and found that only 15% of the keywords were still being censored in 2021. The keywords that remained in the censors' lists were all words pertaining to topics that are permanently controversial in China, examples of which were discussed in Section 2.4.

Only 8% of the keywords in the CitizenLab's chat application list are being censored by the GFW's HTTP censors, indicating that keyword lists by major chat applications and censors in network infrastructure are maintained separately and do not contain much overlap [70]. Unfortunately Rambert et al. do not report on where these differences stem from or what kind of keywords are in the overlap. It would be interesting to know how the censored topics differ between the two areas.

The HTTP censors seem to miss requests around 25% of the time likely due to hitting their computational limits [70]. This makes the GFW's HTTP censorship system considerably less robust than its DNS censorship system which has around 99% success rate under normal constrains [47].

The GFW only censors HTTP requests, and HTTP responses are not being censored [70] [65]. However, in the early 2000's, HTTP censorship in the GFW also included censorship of HTTP responses by keywords in the actual website data [94]. According to Park et al. [65] the GFW's HTTP response censorship was discontinued somewhere between years 2008 and 2009. This was likely due to the GFW of the time being able to censor HTTP responses only around 51% of the time since the RST packets it injected often reached the client too late.

4.2 HTTPS censorship

The GFW censors HTTPS requests by looking for blocked domains in the TLS SNI extension header in ClientHello messages [20] [23]. The censor severs unwanted connections through a TCP reset attack by sending RST packets to both the client and the server. This means China's HTTPS censorship follows the overall model described in Section 2.2, which is also globally the most common way to censor HTTPS connections [54].

While China censors HTTPS in a very common way, a recent study has revealed further details about the architecture of the GFW's HTTPS censors. In 2021 Bock et al. [20] discovered that the GFW has two types of on-path HTTPS censors that are colocated and work together as a pair in a redundant manner. This means that instead of multiple HTTPS censors it might more accurate to say the GFW has multiple pairs of HTTPS censors.

One censor is a primary censor while the other one seems to act as a backup for the primary one [20]. The primary censor is the one that under normal circumstances inserts

TCP RST packets after detecting an unwanted domain. The backup censor does not act at all if it notices the primary censor successfully reacted to the ClientHello. However, if the primary censor for some reason fails to inject the RST packets, the backup censor tries to save the situation by injecting one TCP RST packet after it observes the second client-side packet in the TLS handshake, the ClientKeyExchange. Bock et al. inferred that the censors might not internally share any state information but instead the backup censor simply decides to either inject RST packets into the connection or discard its TCB depending on whether it observed on the line the RST packets injected by the primary censor.

The GFW's HTTPS censors successfully block requests around 97% of the time [20] [84]. Bock et al. inferred this to be the probability of either one of the HTTPS censors blocking the connection. Both the primary and the secondary censor block the same domains, bi-directionally, and on all TCP ports.

Whether HTTPS requests trigger residual censorship and for how long seems to have changed with time: In 2017 Wang et al. [84] reported HTTPS requests triggering residual censorship for 90 seconds, but in 2019 Chai et al. [23] reported this to have changed to 60 seconds. In 2021 Bock et al. [20] recorded no residual censorship of HTTPS requests. However, in 2024, Hoang et al. [48] reported residual censorship affecting both HTTP and HTTPS connections. It is unclear whether these inconsistencies reflect some unknown property of the GFW or simply reflect changes made to it by its operators.

In order to gain insight on the number and categories of domains under HTTP and HTTPS censorship in China, let us take a look at three recent studies. In 2024 Hoang et al. [46] made an extensive study focusing on the HTTP and HTTPS censorship of the GFW. According to their results, out of as many as 1.02 billion domains tested, the GFW censors 943 000 pay-level domains by HTTPS censorship and 55 000 by HTTP censorship (as an example, helsinki.fi is a pay-level domain). This is likely the largest number of domains tested and discovered in research of the GFW's web traffic censorship. Following their previous work with GFWatch [47], Hoang et al. developed a similar longitudinal measurement platform called the GFWeb that measures the GFW's HTTP and HTTPS censorship by testing domains daily and reporting the domains' status as censored or uncensored [46]. They formed the base pool in the same way as they did in their previous study on the GFW's DNS censorship [47], which makes the numbers between the two studies quite comparable. Based on these results, the GFW censors three times more domains through HTTPS censorship as compared to DNS censorship. However, Hoang et

al. also tested twice as many domains against the HTTPS censors as compared to DNS censors, which very likely increases the number of discovered censored domains. It still seems likely HTTPS censorship covers more domains than DNS censorship.

There are also two smaller studies from recent years that have focused on HTTP and HTTPS censorship in China. Chai et al. studied China's HTTPS censorship in 2019 and report that 21 446 sites out of Alexa's top 1 million domains at the time were blocked by the GFW's HTTPS censorship [23]. They conducted their tests between one VPS in China and one in the US. Jin et al. report that in 2021 China censored 22.1% of their tested domains with HTTP keyword filtering and 24.1% with HTTPS censorship [54]. The base pool they used was fairly small, between 1908 to 3953 domains, and it was the same they tested DNS censorship with. These three studies do not agree on the top censored categories, and the categorisation results seem to be largely domain pool dependent.

All three studies agree that the domain or keyword lists used in HTTP, HTTPS, and/or DNS censorship are not the same, but have some overlap [23] [54] [46]. All studies then again give wildly different results regarding the sizes of overlaps between different methods, so the only conclusion to be drawn is that the methods share some blocked domains and that the censored domain or keyword lists are extremely likely maintained separately for different components of the GFW. Policy consistency is interesting from a circumvention point of view, since if a country censors domains inconsistently between different methods, it increases the effectiveness of using protocol specific circumvention methods that defeat only one type of censor [54], such as encrypted DNS.

Since we have discussed HTTP and HTTPS censorship, let us also briefly take a look at HTTP/3 from the point of view of Internet censorship. HTTP/3 [14] is a new version of HTTP that uses QUIC [52] as its underlying protocol, which in turn works on top of UDP. HTTP/3 has been seen as promising from the point of view of Internet censorship resistance due to its built-in encryption and faster connection set up times, which tend to be negatively affected by censorship circumvention tools [37].

In 2021, Elmenhorst et al. [37] empirically evaluated how HTTP/3 and QUIC were being censored in China, India, Iran and Kazakhstan. Their results show that HTTP/3 requests were less frequently censored than HTTPS requests and at the time they did not find proof of any countries specifically targeting HTTP/3 with censorship. This is likely due to HTTP/3 being a very new protocol that has not been widely deployed yet. IP address blocking of servers in China and India affects HTTP/3 traffic the same way it affects HTTPS, however, this is not censorship specifically targeted at HTTP/3 or QUIC.

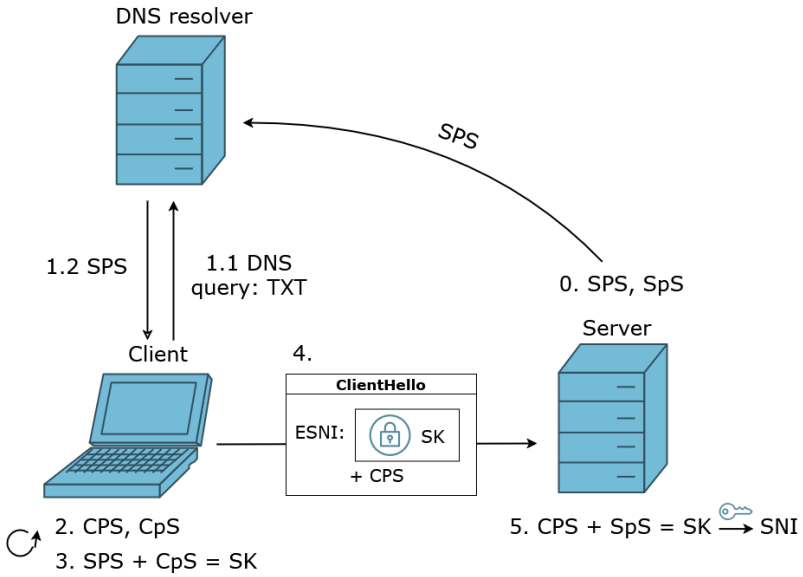

Figure 4.1: Encrypting the ESNI header in TLS ClientHello. Abbreviations in the figure stand for: SPS = server public share, SpS = server private share, CPS = client public share, CpS = client private share, SK = symmetrical key.

Elmenhorst et al. point out that if a protocol such as HTTP/3 is seen as threatening from the censor's point of view, it is possible for a country in the early stages of adoption to completely block all traffic using said protocol without significant collateral damage. China has done this in the past with other privacy enhancing protocols [19].

4.3 Circumventing web traffic censorship: Encrypted Server Name Indication

The problems caused by the plaintext SNI header are well known, and methods for encrypting the server name indication have been developed to enhance the privacy of HTTPS connections. The two most prominent methods are two proposed extensions to TLS 1.3 known as Encrypted Server Name Indication (ESNI) and Encrypted Client Hello (ECH). ESNI and ECH are actually an earlier and a finalized version of the same work-in-progress Internet-Draft (RFC) [73]. Since ESNI is an earlier version of the method, its implementations have been studied more extensively in Internet censorship context and we will concentrate on ESNI in this chapter, even though ECH is the finalized version that will become an actual TLS extension. From Internet censorship point of view ESNI and ECH

provide the same effect - they encrypt the domain name of the target web server in the ClientHello. Both ESNI and ECH add a new extension header into the ClientHello that contains the domain name the client is trying to reach but in encrypted form. In ESNI the header is called the ESNI header. The original SNI header can contain a dummy value or be empty. The main difference between ESNI and ECH is that ESNI encrypts only the domain name in the ESNI header where as ECH encrypts also other information. From here on we talk about ESNI.

In order to encrypt the ESNI header, the client and server need sufficient keys, but at the beginning of the handshake key information has not yet been exchanged. ESNI adds an additional Diffie-Hellman key exchange [59] to the start of the TLS handshake to solve this problem [73]. The symmetrical key created with this Diffie-Hellman key exchange is then used to encrypt the ESNI header in the ClientHello. The keys negotiated here are only used to encrypt the ESNI header and the normal TLS key exchange follows after this.

The steps in encrypting the ESNI header are depicted in Figure 4.1. In order to prepare for ESNI connections, the server first generates a pair of public-private shares and adds the public share to its DNS record (step 0) [73] [87]. Now, when a client makes a DNS query for the server's A or AAAA record, it also makes a query for the server's public share through a TXT record (step 1). The client generates a pair of public-private shares (step 2), and combines the generated private share with the server's public share to create a symmetrical key (step 3). With this symmetrical key, the client encrypts the value of the ESNI header and sends the ClientHello to the server together with the client's public share (step 4). The server then uses the client's public share and provided cipher information to derive the same symmetrical key which it uses to decrypt the domain name from the ESNI header (step 5). Rest of the handshake continues much like in regular TLS.

Since this process is reliant on a DNS query, it is meaningless to use ESNI, unless the DNS query for the server's domain is also encrypted. This means encrypted DNS, DoT or DoH, must be used for the DNS query in order for the domain name of the server to be truly invisible to censors. This also means that attacks that work against the DNS infrastructure also impact ESNI. If a user wants to use ESNI to circumvent censorship, they must use it for all of their connections, since otherwise it becomes feasible for the censor to block all ESNI traffic without any significant collateral damage.

In addition to studying the role of HTTPS censorship in China, Chai et al. also studied what impact ESNI could have on HTTPS censorship in the country [23]. According to their results 84.5% (39 787) of all censored domains they tested were directly blocked

by IP address, indicating that most sites would remain censored even after DNS and SNI censorship had been circumvented. Through ESNI they were able to unblock only 66 sites, meaning only 66 sites were simultaneously not blocked by IP address, had ESNI enabled and were targets of HTTPS censorship. This means that at the time of writing ESNI could not be used to circumvent HTTPS censorship in China in any meaningful way.

With sampled testing, Chai et al. noticed that many of the IP address blocked sites seemed innocuous but were cohosted together with sensitive sites, suggesting that a large number of the blocked sites were collateral damage [23]. Most studies from recent years have not concentrated on IP address blocking in the GFW, so the results from Chai et al. are quite informative and indicate that the GFW still strongly relies on its IP address blocking system to censor the Internet. The authors note how these results highlight the importance of VPN based circumvention methods that completely hide the destination IP address.

At the time of the study support for the ESNI extension was limited, as out of all major cloud providers and browsers only Cloudflare and Firefox supported ESNI [23]. Chai et al. also note that many Chinese users use browsers modified by local Chinese companies that reportedly self-censor. ESNI or ECH are very unlikely to be included in these browsers, and this can further hinder the adoption in China. In 2019, none of the 14 countries they studied, including China, blocked ESNI.

Not long after, the situation changed in China. In 2020 Bock et al. from GFW Report reported how China had suddenly started to block all connections that used ESNI [19]. This means that the operators of the GFW quickly reached the conclusion that completely blocking ESNI does not cause collateral damage they are not willing to take. If a censor in the GFW observes a ClientHello with ESNI enabled, it puts residual censorship in place and drops the packet, making ESNI censors in-path censors. ESNI censors are a separate new component to the GFW, and the HTTPS censors do not seem to be the ones that impose ESNI censorship [20]. Similarly to HTTPS censorship, ESNI censorship happens on all TCP ports[19]. While Bock et al. report the censorship to be bidirectional, in the same year another study by Alice et al. recorded ESNI censorship to have changed to unidirectional, meaning only requests from China are censored [6]. At least in 2020 ECH was not yet censored in China [19]. How ESNI or ECH censorship evolves in China remains to be seen.

4.4 Circumventing web traffic censorship: TCP- and TLS-based methods

Methods that utilize features of the underlying protocols have also been studied and used in circumvention of HTTPS censorship. One such approach is called TCP fragmentation. The basic idea in TCP fragmentation is to utilize the feature of TCP that allows TCP segments to be fragmented into multiple segments at the sending side and to be reconstructed again at the receiving side. On-path censors that utilize TCP reset attacks maintain TCP state information for connections they monitor but do not always correctly implement TCP segment reconstruction [63] [19] [17]. Thus fragmenting a TCP segment to multiple segments might confuse the censor and cause it to miss the SNI header. The GFW is known to have mechanisms for TCP segment reassembly that do not work perfectly [63] [19] [17]. TCP fragmentation can work against censorship of any protocol that runs on top of TCP, such as HTTP, HTTPS or DNS over TCP, but since HTTPS is arguably the most prominent of such protocols, we discuss TCP fragmentation here in the context of HTTPS censorship circumvention.

Utilizing TCP fragmentation in practice means finding the points at which to fragment the TCP segment so that a request that would otherwise be censored does not trigger the censor [19] [20]. As an example, in the case of HTTPS censorship, one simple fragmentation tactic would be to fragment the segment so that the SNI header is either split into two fragments or pushed out of the first fragment, and to possibly send the fragments in reverse order. Since trying different combinations of tactics can be a lot of manual work, Bock et al. [18] created a genetic algorithm tool called Geneva which learns TCP related circumvention tactics that work for the network location the tool is being run from and automatically applies the tactics. Using Geneva, Bock et al. [19] [20] found and demonstrated successful TCP fragmentation tactics against both the HTTPS and ESNI censors of the GFW.

A new study from 2024 by Hoang et al. [46] shows that the GFW has recently fixed some previous deficiencies it had in TCP fragment reassembly and the GFW currently correctly reassembles out-of-order TCP fragments and is able to correctly read and block the SNI header from such segments. Hoang et al. hypothesise that this is the GFW's answer to automated tools such as Geneva, and the usage of such tools will likely become more difficult in the future.

A very similar new method called TLS record fragmentation was proposed by Niere et

al. in 2023 for HTTPS censorship circumvention [63]. In addition to fragmenting TCP segments it is also possible to fragment TLS records. The basic idea is to force a TLS message to be fragmented among multiple TLS records that are still encapsulated within one TCP segment, or possibly within multiple TCP segments. The goal is to force the SNI header out of the first TLS record and cause the censor to miss it.

Niere et al. [63] successfully circumvent the GFW by using TLS record fragmentation only and also by combining TLS and TCP fragmentation. The authors demonstrate that while the GFW has mechanisms to counter TCP fragmentation, it does not currently account for TLS record fragmentation. Compared to TCP, TLS is a higher level protocol in the TCP/IP stack which means manipulating TLS related elements often requires less privileges on a system and this might make TLS record fragmentation easier to implement in practice. Niere et al. also note that TCP and TLS based circumvention techniques are interesting from the point of view of QUIC which combines elements from both protocols to work on top of UDP. Time will tell whether TLS record fragmentation will establish itself as a staple circumvention method much like TCP fragmentation.

Another circumvention method that can work against stateful censors is called TCP state confusion [84]. TCP state confusion is a method that can work against stateful censors, so all censors that maintain state information through TCBs for TCP connections they monitor. The basic idea in TCP state confusion is to send TCP segments to the censor that cause the censor's TCB to become desynchronized from the client and server's TCBs [84] [17]. If the censor updates its state differently from the client and the server, the censor likely ignores following packets in the connection as defective, and the client and the server can communicate freely. Similarly to TCP fragmentation, TCP state confusion can be used against censorship of any protocol that runs on top of TCP, but since HTTP and HTTPS are the most prominent protocols affected, we cover TCP state confusion in the context of HTTP and HTTPS censorship circumvention.

Since a single censor can only reflect one TCP implementation at a time, it is possible to exploit corner cases that stem from differences between TCP implementations to purposefully craft packets that desynchronize the censor [84]. The inserted segments can have varying attributes that break the protocol on purpose such as out-of-window sequence numbers, incorrect checksums, duplicate packets or packets unexpected in that state of a connection. One example that can work against the GFW, is to insert a RST or FIN packet early in the connection which causes the censor to discard its TCB for that connection. When inserting segments from the client side the purpose is that the segments

only affect the censor but are ignored by the server. This can be achieved for example by manipulating the Time To Live (TTL) values in the packets so that the segments reach the censor but not the server, or by inserting a wrong checksum, which would cause the server to discard the packet but not the censor. In case of the GFW, the checksum tactic has been recorded to work, since the censors do not validate the checksum unlike the end points do.

Likely the biggest challenge in utilizing TCP state confusion against censors in practice is the heterogeneity of the network [84]: In addition to censors of the GFW, ISPs can have their own middleboxes that manipulate packets in unexpected ways. Censors of the GFW that censor the same protocol can also behave slightly differently from each other due to differences in software versions. In addition to this, TCP state confusion tactics that work against censors of one protocol do not necessarily work against censors of other protocols [17]. These types of variations in the network can cause TCP state confusion strategies to work differently at different times and in different network locations. Even without accounting for the heterogeneous network landscape, TCP state confusion relies on finding differences in TCP implementations, which often requires large amounts of meticulous work.

Thus, the only way to practically find successful TCP state confusion tactics is to automate the process. Wang et al. [84] studied TCP state confusion tactics against the GFW's HTTP censors in 2017 and in the process developed a tool called INTANG that automatically attempts to find TCP state confusion tactics that best work for that particular network location and target server. They find four new tactics that work against HTTP censors and the combined average success rate for accessing censored content over HTTP from their vantage points within China when using INTANG is 98.3%, which is very high.

Similarly Bock et al. [17] used Geneva [18] to find TCP state confusion tactics in four different countries, China included. Their focus was on finding tactics that can be run completely from the server-side without requiring any client participation, which could improve the ease and safety of using circumvention tools in high censoring countries. They list eight different TCP manipulation sequences that work against the GFW, however the success rates of different tactics and protocols wary greatly: the highest reported success rate against HTTP censors is 54%, against HTTPS censors 55%, and against DNS censors 89% with some tactics practically not working in China. It is interesting to note that while they searched for server-side circumvention methods for five different protocols, DNS, File

Transfer Protocol (FTP), HTTP, HTTPS and Simple Mail Transfer Protocol (SMTP) in four different countries, China, India, Iran and Kazakhstan, China was the only country that censored all five protocols, and it was considerably more difficult to find successful tactics there compared to the other countries.

5 Censorship of circumvention tools in the Great Firewall

Packet filtering systems such as the Great Firewall censor common protocols such as DNS, HTTP, and HTTPS through different types of censors. In order to efficiently evade national packet filtering systems, researchers and developers have created protocols and software that aim to defeat censorship of any protocol all at once. The goal of such generic censorship circumvention tools is to completely hide both the destination domain name and IP address as well as all message content exchanged in sensitive connections. Additionally, these circumvention tools try to make their traffic flows appear as invisible to the national censors as possible to avoid drawing attention. The terms circumvention software and circumvention tool are used interchangeably and both refer to the software implementation of a circumvention protocol. Censorship circumvention tools share many attributes with Virtual Private Networks (VPNs) but their goal is more focused: VPNs offer general privacy and also hide the destination IP address, but the purpose of circumvention tools is to specifically bypass national packet filtering systems which often imposes additional difficulties.

In this chapter, we discuss generic circumvention tools as well as how the Great Firewall tries to prevent their usage. In Section 5.1 we introduce the basic concepts that circumvention protocols rely on and what methods censors have developed to identify and block said protocols. Since both circumvention tools and censorship systems around the world are actively being developed, there is an ongoing cat-and-mouse game between the censors and circumvention tool developers where advances on one side drive the development of the other. To better illustrate this in the context of the Great Firewall, in Section 5.2 we take a closer look at how two famous circumvention protocols, Tor [79] and Shadowsocks [76], have been blocked in China. Section 5.3 details a relatively new development in the Great Firewall: how the Great Firewall managed for a period of time to efficiently and completely passively censor multiple fully encrypted circumvention protocols where it previously needed considerable amount of more complicated active work to achieve the same. In Section 5.4 we briefly discuss some known tactics for making circumvention protocols more resistant to censorship.

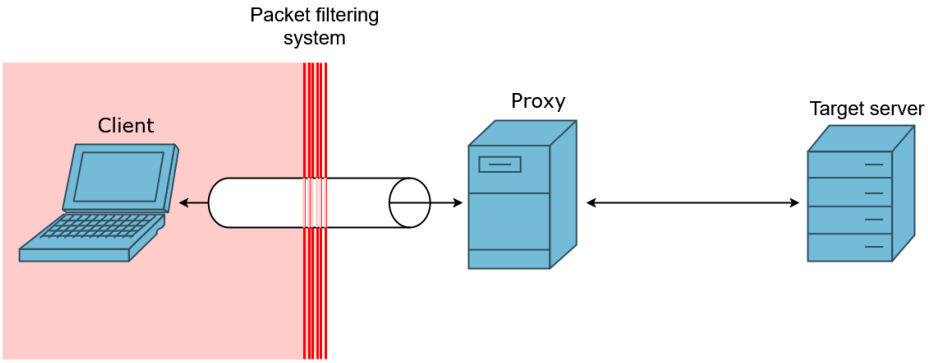

Figure 5.1: Proxying requests over a packet filtering system. The client forms a secure tunnel with the proxy located outside of the censoring regime and accesses the target server through the proxy.

5.1 Generic circumvention tools and their censorship

Many censorship circumvention tools are proxies [33]. Figure 5.1 depicts the basic principle of proxying requests over a national packet filtering system: Proxy servers running the circumvention software are located outside of the censoring regime and users can install a client that enables them to connect to the proxy. The goal is to establish a secure tunnel between the client and the proxy, so that the packet filtering system in between them, such as the GFW, does not react to packets sent in the connection. After the tunnel has been established, the client securely transfers a request, such as HTTPS request, to the proxy. When packets from the client pass through the packet filtering system, the destination IP address that is visible to the censor is the IP address of the proxy and not the target server. The IP address of the target server and the request are encapsulated in the secure tunnel and are not visible to the censor. After receiving a request, the proxy forwards the request to the actual target server and returns the target server's reply back to the client through the secure tunnel.

Different circumvention tools have taken different approaches in creating the secure tunnel and hiding their traffic from the censors, but overall the approaches can be divided into two broad categories: a circumvention tool either tries to make the traffic seem equivalent to completely random noise or exactly like innocent, non-circumventing traffic. The idea is that if the censors cannot confidently tell circumventing traffic apart from allowed traffic, they cannot block the circumventing traffic without risking too much collateral damage.

In order to understand the methods circumvention tools use to create the secure tunnel, it

is necessary to understand what the censors are looking for when they are trying to detect and subsequently block traffic originating from circumvention tools: **Fingerprinting** is the act of trying to detect unique features of a service, in this case features of circumvention traffic [33]. Censors try to detect some unique pattern or attribute of traffic originating from a circumvention tool that can be used to differentiate it from allowed traffic with sufficient confidence. In principle all and any information can be used as a basis for fingerprinting: patterns in values exposed in packet headers, statistical packet information such as packet lengths, payload byte entropy, or communication patterns are few examples [91] [6]. Byte entropy measures the randomness of a byte: if a payload's byte entropy is high, it means the fraction of 1s and 0s in each byte is close to 50/50, meaning the payload resembles random data. Some fingerprints can be undeniable signs that the traffic is what the censor suspects it to be while others act as indications for the censor to gauge further information to confirm its suspicions. While fingerprinting is used by censors to try to identify circumvention traffic, circumvention tool developers or researchers can also fingerprint the censors by inspecting packets injected by the censors. Information gained this way can be used in the development of circumvention software.

Following the classification by Dixon et al. [33], methods that circumvention tools use to create the secure tunnel can be categorized to the following three groups: randomization, mimicry, and tunnelling. All methods aim to make it difficult for censors to fingerprint the traffic. Let us note that conventional encryption schemes, such as using TLS to encrypt the application payload of HTTP traffic, naturally help in hiding the data that is being exchanged, and encryption is an essential part of any circumvention software. However, conventional encryption schemes tend to expose metadata about the connection through plaintext handshakes or unencrypted headers that are full of information the censors can use, such as the target server's destination IP address [33]. Circumvention software needs to cover these information leaks and more in order to circumvent aggressive and well-resourced censors such as the GFW.

Randomization based traffic obfuscation techniques aim to hide all fingerprintable information by having all bytes in the connection, starting from the first byte of the first data packet exchanged after the TCP handshake, to appear completely random [33]. Randomized traffic is also known as fully encrypted traffic [91]. The basic idea is to encrypt the traffic and then further obfuscate it to hide any plain text bytes and statistical patterns. Obfuscation here refers to hiding patterns from the message for example by XOR:ing the message with a known string, and the process is reversible to anyone who knows the mech-

anism used: it is not the same as encryption. The goal is to appear thoroughly random in order to make it difficult to distinguish which protocol is being used.

In order to encrypt the data flow, the client and the proxy need to create encryption keys. The actual data is encrypted with a symmetrical key, which needs to be negotiated first. For this, the client and the proxy use a Diffie-Hellman key exchange. In the Diffie-Hellman key exchange protocol both the client and the server (which in this context is the proxy) generate a private share, that they keep for themselves, and a public share, that they send to each other [59]. Both parties use their private share and the other side's public share to create a symmetrical key.

The challenge that randomization based circumvention tools face is that this initial key exchange also needs to look like random data. For this reason such circumvention tools modify the Diffie-Hellman key exchange so that the public shares are not exchanged as is but are obfuscated with out-of-band information and the messages include random junk or padded data to further obfuscate any patterns [7] [90].

After the client and the proxy have established a shared key, the symmetrically encrypted communication is also further obfuscated to fully randomize it, since it is still possible to derive statistical patterns from conventionally encrypted traffic, which could expose protocol information [33]. The messages can be further XOR:ed with pseudorandom data, and packet sizes as well as the timing of when the messages are sent can be randomized.

Randomized protocols are believed to be difficult for censors to efficiently fingerprint since all traffic essentially looks like network noise. However, randomized traffic has a distinct fingerprintable feature, which is precisely the randomness of the traffic. Most normal web traffic starts with plain text bytes, so the high byte entropy even in the first data packet sets randomized traffic apart from normal traffic. This can and has been used by censors to detect and block randomized circumvention traffic [91].

Examples of censorship circumvention protocols that use randomization are Shadowsocks [76], ScrambleSuite [90] and obfs4 [7]. Shadowsocks is a popular proxy circumvention protocol that utilizes randomization to secure its traffic, meaning its working principle quite closely follows the model described above. ScrambleSuite and obfs4 are more specifically obfuscation protocols that integrate with other protocols to create a secure tunnel through national packet filtering systems. Both work on top of TCP and use randomization to hide the application layer traffic they are carrying. Even though ScrambleSuite and in principle obfs4 as well are protocol independent and can be used to obfuscate many application layer protocols, both are commonly used to obfuscate traffic from an anonymity

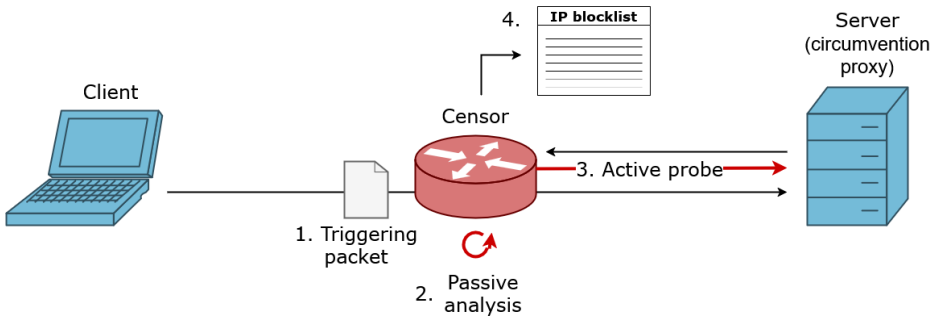

Figure 5.2: Passive analysis and active probing. The censor notices a suspicious connection through passive analysis and actively probes the server to gain more information. If the server confirms the censor's suspicions and reveals itself as a circumvention proxy, the censor can block the IP address of the proxy.

tool called Tor.

Tor [79] is an anonymity network, and it is often used for censorship circumvention despite it technically not being a circumvention tool. Tor consist of a network of encrypted relays that are operated by volunteers [3] [89]. By routing traffic through this network, Tor hides the client's IP address from the target server, and the target server's IP address from anyone inspecting the traffic before it leaves the network. Tor has a list of 'main entrance' relays, also called public bridges, that are used to access the network, but in countries with high censorship, these are usually blocked by IP addresses. To allow censored users access to Tor there exist hidden private bridges that can be used to access the network if the public bridges are not available. The idea is that it is easy for anyone to know a handful of private bridges, but very difficult for someone to know all of them. Bridges can run obfuscation protocols to create a secure tunnel through which clients can connect to them and to make it less likely for censors to find and subsequently block the bridges [38] [36].

The remaining two traffic obfuscation methods are mimicry and tunnelling. In mimicry the goal is to make the circumventing traffic appear as normal traffic from a target protocol by for example prepending an innocent looking target protocol header into the packet to fool the censor [33]. The target protocol can be for example HTTP or DNS. The challenge in mimicking other protocols is that circumvention tools rarely implement the whole target protocol and thus often fail to follow the semantics of it, which provides a way for censors to fingerprint mimicking circumvention traffic from normal traffic of the target protocol.

In tunnelling the idea is to leverage an uncensored protocol and hide circumvention traffic

inside it as a payload [33]. Circumvention traffic has been tunnelled for example in WebRTC [86] streams instead of video content [11] and even real-time video game traffic has been studied as a potential carrier for circumvention traffic [45]. The goal in tunnelling is, similarly to mimicking, to ensure that the tunnelled circumvention traffic believably looks like normal traffic of the tunnelling protocol.

Now that we know what circumvention tools are trying to accomplish and how, let us take a look at what methods the censors use to try to identify and prevent the usage of these tools: The act of censors analysing packets passing through them to look for fingerprints of circumventing traffic is more specifically called **passive analysis** [90]. If a censor recognises a fingerprint during passive analysis, it can try to confirm its suspicions by **actively probing** the (proxy) server the client is accessing [90] [6] [91]. This passive-active combination method is depicted in Figure 5.2. Through the active probes the censor either tries to simulate a genuine connection or to otherwise elicit a response from the server that would enable the censor to confirm the server is a proxy running a circumvention service. If the proxy accepts the connection or otherwise confirms the censor's suspicions, the censor can render the proxy unusable by blacklisting its IP address. After that, all further connection attempts from clients to that proxy will be efficiently blocked based on the proxy's IP address in the destination IP address field and the proxy will become unusable to clients from within that censoring regime. IP address blocking is unlikely to cause extensive collateral damage in this scenario since circumvention proxies are rarely cohosted together with non-circumvention services [6].

In principle, censors can also probe suspicious servers without any initial trigger from passive analysis [38]. However, since active probing can be used to verify the results of passive analysis and passive analysis helps in targeting probes to suspicious servers, the tactics complement each other well and are commonly used together. Circumvention tools often contain functionality to defend against active probing.

5.2 Passive analysis and active probing

The GFW has censors that perform passive analysis as well as probers that perform active probing [6] [91]. The GFW has had active probers at least since 2011 [88], and both the passive analysis and the active probing systems of the GFW have evolved greatly over time. In this section we take a look at two examples of how the Great Firewall has censored circumvention protocols, Tor and Shadowsocks, through the combination of

passive analysis and active probing.

Censorship of Tor

Based on old posts on the Tor website the GFW has targeted Tor at least since 2008. In the beginning the censorship was based on simpler methods such as blocking access to the main website of the Tor project to make it more difficult to gain information about the tool and access the network [67]. A year later the main entrance relays were IP address blocked in China [80], and in 2011 the GFW started to dynamically use passive analysis together with active probing to also block Tor's hidden bridges [89].

The Chinese censors managed to fingerprint Tor by the TLS cipher list that the Tor client sent in its Client Hello message when connecting to a bridge [89] [88]. The cipher list was previously identical to the one used by Firefox, but it had then become something only Tor used, enabling the censors to mark all connections that used this TLS cipher list as suspicious during passive analysis. After a connection was marked, the censors probed the bridge the client was accessing and similarly tried to make a Tor connection to it. The probes seemed to happen at 15 minute intervals, which suggested that the GFW used some form of probing queue that was processed every 15 minutes. If the bridge accepted the Tor connection from the prober, the censor blacklisted the IP address and port combination of the bridge, making Chinese users unable to use that bridge. The bridge could be blocked within minutes of the first user connecting to it [38]. After this incident circumvention tool developers were able to make Tor more censorship resistant by deploying and using obfuscation protocols.

According to Winter and Lindskog [89] the active probes came from a large pool of IP addresses that the GFW had spoofed from the largest ASes in China. Three years later Ensafi et al. [38] studied the GFW's active probing mechanism in the context of Tor censorship and deduced that even though the IP addresses in the probes vary greatly, the probes seem to originate only from few processes [38]. From this they hypothesise that the active probing system is actually a network of distributed proxies that forward probe packets from a centrally controlled system. Ensafi et al. gained this insight by fingerprinting the active probers based on patterns they observed in multiple TCP fields. It also seems the 15 minute queue system from 2012 had been replaced by a near real time probing mechanism by the time of the study, since in their tests most of the probes now arrived after under a second from the first test user connecting to their bridge.

Whether different Tor integrated obfuscation protocols work in China or not evolves with time as older protocols become unusable and new ones are developed. The latest Tor related obfuscation protocol research seems to have been done in 2018 by Dunna et al. [36], but it is difficult to say whether the provided results are still accurate today.

Censorship of Shadowsocks

The Shadowsocks protocol has intermittently been blocked by the GFW multiple times. Alice et al. [6] studied in detail one such period of blockage Shadowsocks experienced in 2019. In order to narrow down what attributes the GFW used as signs of Shadowsocks traffic during passive analysis, they sent random data of varied entropies and lengths though the GFW towards their test Shadowsocks proxies and observed which packets triggered active probes from the GFW. Alice et al. discovered that the GFW examined the byte entropy and length of the first data packet sent after the TCP handshake. Since Shadowsocks is a fully encrypted protocol, even the first packet in the Shadowsocks TCP connection has high byte entropy. In addition to this, the length distribution of Shadowsocks packets resembled the distribution of the traffic it carried, typically HTTPS, since Shadowsocks did not properly pad the obfuscated packets but only added a constant number of bytes to it. The GFW was able to use the high byte entropy and HTTPS typical packet length as an indication that the traffic might be Shadowsocks traffic. Next, in order to confirm its suspicions, the GFW actively probed the server (proxy) the client was accessing.

In addition to discovering the fingerprints used by the GFW, Alice et al. also thoroughly analysed the active probes their test proxies received [6]. The GFW could send three different types of probes to the suspected Shadowsocks proxy: probes consisting of random data of various lengths, exact replays of previous successful connections, or replays of first packets of previous connections with some of the original bytes manipulated in tactical places. Tactical places here mean bytes where for example different encryption related values negotiated in the beginning of the Shadowsocks connection are placed. Some Shadowsocks proxies block exact replays of previous connections, so manipulating bytes in a way that the resulting message still structurally follows the Shadowsocks protocol can increase the likelihood of the probe being accepted as a new legitimate connection. If the Shadowsocks proxy answered to the probing in a Shadowsocks typical way, the GFW blocked the proxy's IP address. After the IP address was blacklisted the GFW dropped all packets originating from the proxy. Once blacklisted, the proxy could become unblocked later without the GFW sending more probes to confirm whether the server was

still running Shadowsocks.

Similarly to active probes the GFW sent to Tor bridges, the probes to the Shadowsocks proxies came from a large constantly changing pool of IP addresses [6]. It is quite safe to assume the same proxy network is being used to send probes to proxies running different circumvention tools, even when the centralised system behind it needs to have protocol specific functions to properly fingerprint different tools.

Alice et al. note that only few of their Shadowsocks proxies were actually blocked during the study period despite the proxies receiving massive amounts of active probes from the GFW [6]. This lead them to hypothesise that the actual blocking of circumvention servers in the GFW might be human controlled. This hypothesis is further supported by the fact that censorship of the Shadowsocks protocol has been reported to clearly fluctuate with the surrounding political climate in China. Users in China have reported their Shadowsocks servers becoming blocked especially close to politically sensitive events, such as important congress meetings of the Chinese Communist Party (CCP) or the anniversary of the Tiananmen square massacre. After such sensitive events were over users in China reported their Shadowsocks proxies becoming unblocked again. This is an interesting example of how Internet censorship in China combines automated and manual elements, and also how the surrounding political events affect the degree of censorship.

Alice et al. disclosed their findings to the developers of the two Shadowsocks implementations they used in their tests and at least one implementation made modifications to their code that successfully enabled their servers to become unblocked in China again [6].

5.3 Fully passive censorship of circumvention tools

In November 2021 there was a new development in the GFW's censorship of circumvention protocols: the GFW was now capable of completely passively detecting and blocking randomization based circumvention traffic in real time [91]. This new type of censorship affected many popular circumvention and obfuscation protocols such as Shadowsocks and obfs4. As was discussed in Section 5.2, the GFW has been passively analysing and actively probing randomized protocols for a long time, but this new censorship system achieved the same or better results without any active probing. The new system seems to be separate from the passive-active censorship system and they work in parallel. Similarly to previous blockings of Shadowsocks, the launch of this new passive censorship system coincided with a politically sensitive event, in this case an important meeting of the Chinese Communist

Party.

Wu et al. [91] studied this new censorship system extensively between years 2021 and 2023. According to their results, the new system passively analyses traffic, and implements efficient heuristics that exempt traffic that is very unlikely fully encrypted circumvention traffic, and blocks everything else. In principle the system whitelists traffic instead of blacklisting, which can work well against randomized protocols [33] [90].

The heuristics use fingerprints of common protocols, byte entropy, and the number, fraction, and position of printable ASCII characters in the first data packet to make its assessment [91]. The first rule is that if the first 3 to 6 bytes of the packet match TLS or HTTP, even if the rest of the packet does not follow the protocol, these connections are exempt. TLS and HTTP are likely explicitly exempt first in the beginning of the system's checklist to rule most traffic out immediately and to do the other checks only to a small fraction of traffic. Secondly, the system allows connections where the first six bytes, half of the characters, or 20 contiguous bytes are printable ASCII. The idea is likely to allow plaintext protocols that are susceptible to other censorship methods. Lastly, if the byte entropy in the packet is sufficiently low, the connection is likely to get exempt. Otherwise, the packet is dropped. Some of the rules overlap, such as the printable ASCII character rules and first bytes of HTTP.

After the first packet has been dropped, the client - server pair is put under residual censorship that lasts for 180 seconds [91]. During the residual censorship all TCP packets are dropped from the client to the server, but not the other way around. The GFW likely limits the number of connections it residually blocks since if multiple connections are initiated the duration of the residual blocks decreases. Likely in order to not over block and to save resources, the GFW only monitors around 26% of all connections with this new system and only IP addresses that belong to AS's that are from popular VPS providers which are often used to host circumvention servers. The 26% rule is likely enough due to the residual censorship that blocks subsequent communication attempts between the same client and the server.

According to Wu et al., the system only affects TCP traffic and all ports can experience blocking [91]. Blocking does not trigger unless the TCP handshake has been fully completed. The GFW waits for 5 minutes after the TCP handshake to see the first TCP packet, and only the first packet is analysed. Wu et al. simulated the heuristics of the GFW in a network where usage of circumvention tools is not expected and observed the rules to block innocent traffic only around 0,6% of the time, which is likely close to the

false positive rate of the new system. In large censorship systems such as the GFW, having a low false positive rate is important, since frequent unintended censorship could likely lead to backlash.

Based on their findings Wu et al. suggested improvements to different circumvention protocols, such as prepending a customisable string of printable ASCII characters to the start of first data packet or adding additional zeros or ones into the packet to skew the byte entropy attributes. In February 2023, the strategies adopted by circumvention tools, including multiple different implementations of Shadowsocks, were still effective in China.

5.4 Censorship resistant circumvention

Censorship based on passive analysis and active probing can be avoided either by evading the passive traffic analysis or by responding discreetly to active probes. It is good practice for censorship circumvention tools to defend in both ways.

In defeating passive analysis, fragmenting the client-side TCP segments can help confuse the censor, since that will change the statistical attributes of the traffic flow such as packet length distribution. In Alice et al.'s tests, servers that had a tool that fragmented the client-side packets received considerably fewer active probes [6]. However, these tools are unfortunately also fingerprintable [6], and Winter et al. [89] point out that packet fragmentation methods have their limits: client side fragmentation, if it successfully evades censorship and does not cause active probing, would need to be enabled on all users in the censoring country, or otherwise the server would be probed and possibly blocked. Fragmentation from the proxy side does not have this issue, but if the censor makes its decisions based on the first client-side packet, this does not provide an answer. On the other hand, the mitigation methods provided by Wu et al. [91] prove that sometimes even simple modifications the first client-side packet can help avoid passive analysis. Circumvention techniques based on mimicry can also efficiently resist whitelisting censorship systems such as the one discussed in Section 5.3, but mimicry can be challenging to implement correctly [33]. Overall, passive analysis can be avoided, if the circumventing traffic does not have fingerprints the censor is able to identify with reasonable confidence.

In defeating active probing, circumvention servers should always reply discreetly and consistently in different error cases to avoid creating unique edge cases that can be used by the censor to fingerprint the proxy by the way it responds to probes [6]. We discuss responses to error cases, since naturally if the proxy accepts the connection without any errors, this

already confirms the censor's suspicions. When the proxy receives a connection attempt it does not understand, it can respond for example by timing out the connection, by aborting the connection with a TCP reset packet, or by closing the connection with a FIN/ACK packet. If the proxy responds differently depending what application errors the probes from the censor trigger, it is possible for the censor to statistically fingerprint the proxy with a large number of probes.

As an example Alice et al. [6] managed to fingerprint different Shadowsocks implementations by simulating a prober and recording the answers from their proxies. Depending on the attributes of the probe, such as length of the salt in the Shadowsocks message or the length of the whole probe, the proxies respond in different ways when the probe does not correctly adhere to the Shadowsock protocol. This makes it possible for the censor to map the thresholds at which probe lengths the proxy changes its behaviour, which creates a Shadowsocks implementation specific pattern, allowing for fingerprinting. There is no clear evidence of censors using this method in practice, but Alice et al. note that the GFW already sends multiple probes of lengths that match the behaviour thresholds. Authenticating the clients and using strong ciphers help in defending against probes that are exact replays of previous connections.

6 Discussion

Challenges in Internet censorship research

While Internet censorship research is often practically the only way to obtain information about governmental Internet censorship, it is also not without its limitations. Domain related results seem to be highly affected by what domains were tested and in what numbers and numerical results between different studies are not easily comparable. Many Internet censorship studies also only reflect the situation in a specific country at the specific moment in time when the study was conducted. High censoring countries tend to update their censorship systems actively and these kinds of studies inevitably become outdated relatively fast. Said studies do serve to provide historical data but they only do so for a single point in time. This leads to a situation, where different research groups repeat similar studies every few years to capture developments in countries of interest.

To counter this long-standing trend, there are some projects whose main focus is to provide longitudinal data either globally or from a specific country and censorship method. Such Internet censorship and interference observatories actively measure and report on the state of Internet freedom in the world while providing researchers with vital data about Internet availability and censored domains worldwide. Some notable projects are Censored Planet [22], OONI [64] [40], The Citizen Lab [78], ICLab [62], Quack [82], and Access Now [4]. A non-governmental organization Freedom House also makes yearly reports on the status of Internet freedom around the world that provide valuable information on the political context of Internet censorship as well as its impacts from a human rights perspective. GFWatch and GFWeb developed by Hoang et al. are examples of long-term observation projects that monitor the GFW specifically [47] [42] [46] [43]. Both projects aim to better provide long-term data on web traffic censorship in China and to better capture developments of the GFW.

Some Internet censorship studies also only measure whether in a technological sense Internet censorship can be observed in a country overall without taking into account the purpose or the actor behind the censor. This is something that needs to be taken into account when drawing conclusions from research results, since in real life the surrounding context of the censorship matters. It is understandable that evaluations of political

context might be out of scope for many research groups consisting mostly of computer scientists, whose focus is on the technical execution of Internet censorship. Reports by Internet rights activist groups or political science studies focus on the intention, effects, and surrounding political context of Internet censorship, but on the contrary lack information about the actual technological implementations. Ultimately, Internet censorship research needs to be a multi-disciplinary effort, since both technical and human aspects need to be taken into account when forming an accurate picture of Internet censorship globally or within a country.

Even though Internet censorship research is often motivated by the will to help citizens in oppressive regimes, in addition to circumventing governmental Internet censorship the developed circumvention methods can also be used to access content for criminal purposes. This might be an inevitable side-effect of greater anonymity on the Internet, but it is a side to censorship circumvention that has not been much discussed, at least not in research context.

Overall, discussions about Internet censorship and its ethics ultimately come down to what kind of censorship is considered acceptable and what oppressive. According to a global survey conducted by the Internet Society in 2013, most people want some level of censorship on the Internet, mostly to make the Internet a safer place [44]. The survey collected answers from more than 10 000 people from 20 countries around the world regarding different aspects of Internet usage. It is evident not all governmental Internet censorship is seen as oppressive. From a research point of view however, it is interesting to concentrate on the negative side of Internet censorship: how governments abuse Internet censorship to manipulate the public opinion and oppress the human rights of their citizens. According to the survey, most people do think freedom of expression should be guaranteed on the Internet and see access to Internet as a basic right.

There have also been some concerns on how most Internet censorship research is concentrated on countries that are known to be offenders [61] [62]. This might lead to blind spots when it comes to keeping up to date with Internet censorship developments in more traditionally liberal countries. For example, relatively little research has been done on Internet censorship in European countries [83] [21] or the USA [81].

While the USA is not an active target of Internet censorship research, it is necessary to mention that multiple companies in the country have had unfortunate ties to governmental Internet censorship in other countries, particularly in China [95]. An especially notorious example is the telecommunications giant Cisco Systems, who built multiple Chinese

Internet backbone networks and helped enable the large-scale censorship of today. Since much of China's Internet backbone at least in the beginning of the century ran on Cisco's routers, they have likely been involved in the active maintenance of the systems and have provided help in configuring the routers for censorship purposes.

Future directions

Circumvention tools that hide both the domain and the IP address of the target server are likely the most efficient methods to circumvent the Great Firewall. It makes it unnecessary to defeat different censors of different protocols separately, and circumvention tools and VPNs are also the only methods that enable circumvention of IP address blocking. However, due to the legislation in China, obtaining or sharing information about, as well as running and using circumvention tools or VPNs has been made purposefully difficult and poses a risk to users. Circumvention tools and VPNs are actively under the watching eye of the government and China has in recent years only tightened its legislation surrounding circumvention tool and VPN usage [25]. This hinders the adoption of such tools and also raises the bar high enough so that only users with sufficient technological knowledge and motivation can circumvent censorship. Circumvention methods that do not require active participation from the user would help level the playfield and nullify the effects of governmental Internet censorship for as many people as possible. Extensions to common protocols that undermine existing censorship methods, such as encrypted DNS, ESNI or ECH, have in this regard the potential to reach mainstream adoption on a different scale from circumvention tools or VPNs, however, their adoption will not go smoothly in high censoring countries such as China.

Overall, it seems unlikely that Internet censorship policies in China will considerably relax in the near future, as there are only signs of opposite developments with China tightening its legislation surrounding Internet censorship [25] and increasing the number of censored domains in recent years [54] [47] [9]. In addition to tightening national control, China has also been exporting their model of state-controlled Internet to questionable authoritarian leaderships in developing countries around the world by providing both networking infrastructure and tools of mass surveillance, as well as training governments in using said infrastructure to enforce digital authoritarianism [29]. According to Freedom House, in 2024 global Internet freedom declined for the 14th consecutive year [41].

7 Conclusions

In this thesis we discussed governmental Internet censorship and how it has been implemented in the national packet filtering system of China, the Great Firewall. Governments around the world utilize diverse methods to control what their citizens can access on the Internet. HTTP and HTTPS censorship are the most used methods around the world to censor web traffic. High censoring countries such as China and Iran also employ DNS censorship to make their web traffic censorship more comprehensive. Since the inner workings of governmental Internet censorship systems are not public knowledge, Internet censorship research aims to uncover in detail the technological methods used in these systems as well as which domains are being censored.

China has consistently ranked among the most aggressive Internet censors in the world. The Great Firewall has DNS, HTTP, and HTTPS censorship components that under normal circumstances manage to censor their target protocols almost every time. The Great Firewall censors an extensive number of domains and blocks access to most large western social media platforms and news sites which the Chinese government cannot control. In addition, the Great Firewall censors a considerable number of domains that host content critical of the government as well as content related to human rights and Internet freedom.

Compared to censorship systems in many other countries, the Great Firewall is not transparent in its censorship and users are not informed when their request has been censored: The DNS censorship system of the Great Firewall returns publicly routable IP addresses that redirect to pages of legitimate businesses that make it difficult to automatically detect when a query has been censored. HTTP and HTTPS traffic is censored through TCP reset attacks without presenting the user with a blockpage indicating the request was censored.

Circumvention protocols have been developed to circumvent governmental Internet censorship and these tools make it possible to bypass DNS, HTTP and HTTPS censorship of the Great Firewall. However, the Great Firewall actively aims to censor circumvention tools through its passive analysis and active probing systems and whether different tools are usable from within China or not varies with time. Sharing information about and providing circumvention services has also been made difficult in China through legislation.

In the field of Internet censorship China is an industry leader that showcases a model of a state controlled national Internet that many other authoritarian countries with similar

ambitions can try to replicate. The different components of the Great Firewall are still being developed and maintained actively and this is expected to continue in the foreseeable future. In order to capture these future developments it is crucial that research and longitudinal measurement projects targeting the Great Firewall continue from here on as well.

References

[1] *1068/2006 | Lainsäädäntö | Finlex*. URL: https://finlex.fi/fi/lainsaadanto/2006/1068 (visited on 03/21/2025).

[2] *A research-oriented top sites ranking hardened against manipulation - Tranco*. URL: https://tranco-list.eu/ (visited on 04/04/2025).

[3] *A short introduction to Tor - Tor Specifications*. URL: https://spec.torproject.org/intro/index.html (visited on 03/28/2025).

[4] *Access Now*. URL: https://www.accessnow.org/ (visited on 03/28/2025).

[5] *Adoption & Usage in China | Cloudflare Radar*. URL: https://radar.cloudflare.com/adoption-and-usage/cn (visited on 03/28/2025).

[6] Alice, Bob, Carol, J. Beznazwy, and A. Houmansadr. "How China Detects and Blocks Shadowsocks". In: *Proceedings of the ACM Internet Measurement Conference*. IMC '20: ACM Internet Measurement Conference. Virtual Event USA: ACM, Oct. 27, 2020, pp. 111–124. ISBN: 978-1-4503-8138-3. DOI: 10.1145/3419394.3423644.

[7] Y. Angel. *GitHub - Yawning/obfs4: The obfourscator (Courtesy mirror)*. URL: https://github.com/Yawning/obfs4/tree/master (visited on 03/23/2025).

[8] Anonymous. "Towards a Comprehensive Picture of the Great Firewall's DNS Censorship". In: *4th USENIX Workshop on Free and Open Communications on the Internet (FOCI 14)*. San Diego, CA: USENIX Association, Aug. 2014.

[9] Anonymous, A. A. Niaki, N. P. Hoang, P. Gill, and A. Houmansadr. "Triplet Censors: Demystifying Great Firewall's DNS Censorship Behavior". In: *10th USENIX Workshop on Free and Open Communications on the Internet (FOCI 20)*. Virtual event: USENIX Association, Aug. 2020.

[10] D. Bamman, B. O'Connor, and N. Smith. "Censorship and deletion practices in Chinese social media". In: *First Monday* (Mar. 4, 2012). ISSN: 1396-0466. DOI: 10.5210/fm.v17i3.3943.

[11] D. Barradas, N. Santos, L. Rodrigues, and V. Nunes. "Poking a Hole in the Wall: Efficient Censorship-Resistant Internet Communications by Parasitizing on WebRTC". In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security. Virtual Event USA: ACM, Oct. 30, 2020, pp. 35–48. ISBN: 978-1-4503-7089-9. DOI: 10.1145/3372297.3417874.

[12] S. Basso. "Measuring DoT/DoH Blocking Using OONI Probe: a Preliminary Study". In: *Proceedings of NDSS Symposium 2021*. Network and Distributed System Security (NDSS) Symposium 2021. Virtual Event, Feb. 21, 2021.

[13] Z. S. Bischof, K. Pitcher, E. Carisimo, A. Meng, R. Bezerra Nunes, R. Padmanabhan, M. E. Roberts, A. C. Snoeren, and A. Dainotti. "Destination Unreachable: Characterizing Internet Outages and Shutdowns". In: *Proceedings of the ACM SIGCOMM 2023 Conference*. ACM SIGCOMM '23: ACM SIGCOMM 2023 Conference. New York NY USA: ACM, Sept. 10, 2023, pp. 608–621. ISBN: 9798400702365. DOI: 10.1145/3603269.3604883.

[14] M. Bishop. *HTTP/3*. Request for Comments RFC 9114. Num Pages: 57. Internet Engineering Task Force, June 2022. DOI: 10.17487/RFC9114.

[15] S. Blake-Wilson, J. Mikkelsen, M. Nyström, D. Hopwood, and T. Wright. *Transport Layer Security (TLS) Extensions*. Request for Comments RFC 3546. Num Pages: 29. Internet Engineering Task Force, June 2003. DOI: 10.17487/RFC3546.

[16] K. Bock, P. Bharadwaj, J. Singh, and D. Levin. "Your Censor is My Censor: Weaponizing Censorship Infrastructure for Availability Attacks". In: *2021 IEEE Security and Privacy Workshops (SPW)*. 2021 IEEE Security and Privacy Workshops (SPW). San Francisco, CA, USA: IEEE, May 2021, pp. 398–409. ISBN: 978-1-6654-3732-5. DOI: 10.1109/SPW53761.2021.00059.

[17] K. Bock, G. Hughey, L.-H. Merino, T. Arya, D. Liscinsky, R. Pogosian, and D. Levin. "Come as You Are: Helping Unmodified Clients Bypass Censorship with Server-side Evasion". In: *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*. SIGCOMM '20: Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication. Virtual Event USA: ACM, July 30, 2020, pp. 586–598. ISBN: 978-1-4503-7955-7. DOI: 10.1145/3387514.3405889.

[18] K. Bock, G. Hughey, X. Qiang, and D. Levin. "Geneva: Evolving Censorship Evasion Strategies". In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. CCS '19. New York, NY, USA: Association for Computing Machinery, Nov. 6, 2019, pp. 2199–2214. ISBN: 978-1-4503-6747-9. DOI: 10.1145/3319535.3363189.

[19] K. Bock, iyouport, Anonymous, L.-H. Merino, D. Fifield, A. Houmansadr, and D. Levin. *Exposing and Circumventing China's Censorship of ESNI*. GFW Report. Aug. 7, 2020. URL: https://gfw.report/blog/gfw_esni_blocking/en/ (visited on 05/04/2025).

[20] K. Bock, G. Naval, K. Reese, and D. Levin. "Even Censors Have a Backup: Examining China's Double HTTPS Censorship Middleboxes". In: *Proceedings of the ACM SIGCOMM 2021 Workshop on Free and Open Communications on the Internet*. FOCI'21. Virtual Event USA: ACM, Aug. 27, 2021, pp. 1–7. ISBN: 978-1-4503-8640-1. DOI: 10.1145/3473604.3474559.

[21] Y. Breindl and J. Wright. "Internet Filtering Trends in Western Liberal Democracies: French and German Regulatory Debates". In: *Proceedings of Free and Open Communications on the Internet (FOCI)*. FOCI'12. Aug. 15, 2012.

[22] *Censored Planet*. Censored Planet. URL: https://censoredplanet.org/ (visited on 03/28/2025).

[23] Z. Chai, A. Ghafari, and A. Houmansadr. "On the Importance of Encrypted-SNI (ESNI) to Censorship Circumvention". In: *Proceedings of the 9th USENIX Workshop on Free and Open Communications on the Internet (FOCI 19)*. FOCI'19. Santa Clara, CA: USENIX Association, Aug. 2019.

[24] *China | OpenNet Initiative*. URL: https://opennet.net/research/profiles/china-including-hong-kong (visited on 03/23/2025).

[25] *China: Freedom on the Net 2024 Country Report*. Freedom House. URL: https://freedomhouse.org/country/china/freedom-net/2024 (visited on 03/23/2025).

[26] *citizenlab/chat-censorship*. original-date: 2014-04-15T18:27:29Z. Apr. 10, 2025. URL: https://github.com/citizenlab/chat-censorship (visited on 04/11/2025).

[27] *citizenlab/test-lists*. original-date: 2014-04-15T19:14:11Z. Mar. 17, 2025. URL: https://github.com/citizenlab/test-lists (visited on 03/23/2025).

[28] R. Clayton, S. J. Murdoch, and R. N. M. Watson. "Ignoring the Great Firewall of China". In: *Privacy Enhancing Technologies*. Ed. by G. Danezis and P. Golle. Red. by D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, and G. Weikum. Vol. 4258. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 20–35. ISBN: 978-3-540-68790-0 978-3-540-68793-1. DOI: 10.1007/11957454_2.

[29] C. Codreanu. "USING AND EXPORTING DIGITAL AUTHORITARIANISM: CHALLENGING BOTH CYBERSPACE AND DEMOCRACIES". In: *SSRN Electronic Journal* (2024). ISSN: 1556-5068. DOI: 10.2139/ssrn.5061337.

[30] *Common Crawl - Open Repository of Web Crawl Data*. URL: https://commoncrawl.org/ (visited on 04/04/2025).

[31] *Complete GFW Rulebook for Wikipedia plus Comprehensive List for Websites, IPs, IMDB and AppStore*. Google Docs. 2014. URL: https://docs.google.com/spreadsheets/u/0/d/11GBNGwMP0X0VKCR5AG1PJ1U2dJCNQG7U3L-Sx8zz1cw/ (visited on 04/11/2025).

[32] 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: *IEEE/ACM Transactions on Networking* 22.6 (Dec. 2014). Conference Name: IEEE/ACM Transactions on Networking, pp. 1964–1977. ISSN: 1558-2566. DOI: 10.1109/TNET.2013.2291244.

[33] L. Dixon, T. Ristenpart, and T. Shrimpton. "Network Traffic Obfuscation and Automated Internet Censorship". In: *IEEE Security & Privacy* 14.6 (Nov. 2016). Conference Name: IEEE Security & Privacy, pp. 43–53. ISSN: 1558-4046. DOI: 10.1109/MSP.2016.121.

[34] *Domain names - concepts and facilities*. Request for Comments RFC 1034. Num Pages: 55. Internet Engineering Task Force, Nov. 1987. DOI: 10.17487/RFC1034.

[35] *Domain names - implementation and specification*. Request for Comments RFC 1035. Num Pages: 55. Internet Engineering Task Force, Nov. 1987. DOI: 10.17487/RFC1035.

[36] A. Dunna, C. O'Brien, and P. Gill. "Analyzing China's Blocking of Unpublished Tor Bridges". In: *Proceedings of the 8th USENIX Workshop on Free and Open Commu-*

nications on the Internet (FOCI 18). Baltimore, MD: USENIX Association, Aug. 2018.

[37] K. Elmenhorst, B. Schütz, N. Aschenbruck, and S. Basso. "Web censorship measurements of HTTP/3 over QUIC". In: *Proceedings of the 21st ACM Internet Measurement Conference*. IMC '21: ACM Internet Measurement Conference. Virtual Event: ACM, Nov. 2, 2021, pp. 276–282. ISBN: 978-1-4503-9129-0. DOI: 10.1145/3487552.3487836.

[38] R. Ensafi, D. Fifield, P. Winter, N. Feamster, N. Weaver, and V. Paxson. "Examining How the Great Firewall Discovers Hidden Circumvention Servers". In: *Proceedings of the 2015 Internet Measurement Conference*. IMC '15: Internet Measurement Conference. Tokyo Japan: ACM, Oct. 28, 2015, pp. 445–458. ISBN: 978-1-4503-3848-6. DOI: 10.1145/2815675.2815690.

[39] R. T. Fielding, M. Nottingham, and J. Reschke. *HTTP/1.1*. Request for Comments RFC 9112. Num Pages: 46. Internet Engineering Task Force, June 2022. DOI: 10.17487/RFC9112.

[40] A. Filasto and J. Appelbaum. "OONI : Open Observatory of Network Interference". In: *Proceedings of Free and Open Communications on the Internet (FOCI)*. FOCI'12. Aug. 8, 2012.

[41] *Freedom on the Net 2024, The Struggle For Trust Online*. Freedom House. Nov. 2024. URL: https://freedomhouse.org/sites/default/files/2024-10/FREEDOM-ON-THE-NET-2024-DIGITAL-BOOKLET.pdf (visited on 03/29/2025).

[42] *GFWatch Dashboard*. URL: https://gfwatch.org/ (visited on 03/23/2025).

[43] *GFWeb Dashboard*. URL: https://gfweb.ca/ (visited on 03/23/2025).

[44] *Global Internet User Survey 2012*. The Internet Society and Redshift Research. Mar. 14, 2013. URL: https://web.archive.org/web/20130314063616/https://www.internetsociety.org/sites/default/files/GIUS2012-GlobalData-Table-20121120_0.pdf (visited on 03/28/2025).

[45] B. Hahn, R. Nithyanand, P. Gill, and R. Johnson. "Games without Frontiers: Investigating Video Games as a Covert Channel". In: *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. 2016 IEEE European Symposium on Security and Privacy (EuroS&P). Mar. 2016, pp. 63–77. DOI: 10.1109/EuroSP.2016.17.

[46] N. P. Hoang, J. Dalek, M. Crete-Nishihata, N. Christin, V. Yegneswaran, M. Polychronakis, and N. Feamster. "GFWeb: Measuring the Great Firewall's Web Censorship at Scale". In: *Proceedings of the 33th USENIX Security Symposium*. 33rd USENIX Security Symposium (USENIX Security 24). Philadelphia, PA: USENIX Association, Aug. 2024, pp. 2617–2633. ISBN: 978-1-939133-44-1.

[47] N. P. Hoang, A. A. Niaki, J. Dalek, J. Knockel, P. Lin, B. Marczak, M. Crete-Nishihata, P. Gill, and M. Polychronakis. "How Great is the Great Firewall? Measuring China's DNS Censorship". In: *Proceedings of the 30th USENIX Security Symposium*. 30th USENIX Security Symposium (USENIX Security 21). USENIX Association, Aug. 2021, pp. 3381–3398. ISBN: 978-1-939133-24-3.

[48] N. P. Hoang, A. A. Niaki, P. Gill, and M. Polychronakis. "Domain name encryption is not enough: privacy leakage via IP-based website fingerprinting". In: *Proceedings on Privacy Enhancing Technologies 2021*. PETS 2021: The 21th Privacy Enhancing Technologies Symposium. Oct. 1, 2021, pp. 420–440. DOI: 10.2478/popets-2021-0078.

[49] P. E. Hoffman and P. McManus. *DNS Queries over HTTPS (DoH)*. Request for Comments RFC 8484. Num Pages: 21. Internet Engineering Task Force, Oct. 2018. DOI: 10.17487/RFC8484.

[50] *HTTPS encryption on the web – Google Transparency Report*. URL: https://transparencyreport.google.com/https/overview?hl=en (visited on 03/28/2025).

[51] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. E. Hoffman. *Specification for DNS over Transport Layer Security (TLS)*. Request for Comments RFC 7858. Num Pages: 19. Internet Engineering Task Force, May 2016. DOI: 10.17487/RFC7858.

[52] J. Iyengar and M. Thomson. *QUIC: A UDP-Based Multiplexed and Secure Transport*. Request for Comments RFC 9000. Num Pages: 151. Internet Engineering Task Force, May 2021. DOI: 10.17487/RFC9000.

[53] L. Jin, S. Hao, H. Wang, and C. Cotton. "Understanding the Impact of Encrypted DNS on Internet Censorship". In: *Proceedings of the Web Conference 2021*. WWW '21: The Web Conference 2021. Ljubljana Slovenia: ACM, Apr. 19, 2021, pp. 484–495. ISBN: 978-1-4503-8312-7. DOI: 10.1145/3442381.3450084.

[54] L. Jin, S. Hao, H. Wang, and C. Cotton. "Understanding the Practices of Global Censorship through Accurate, End-to-End Measurements". In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 5.3 (Dec. 14, 2021), pp. 1–25. ISSN: 2476-1249. DOI: 10.1145/3491055.

[55] *KeepItOn 2024 Internet Shutdowns Annual Report*. Access Now. URL: https://www.accessnow.org/wp-content/uploads/2025/02/KeepItOn-2024-Internet-Shutdowns-Annual-Report.pdf (visited on 03/21/2025).

[56] G. King, J. Pan, and M. E. Roberts. "How Censorship in China Allows Government Criticism but Silences Collective Expression". In: *American Political Science Review* 107.2 (May 2013), pp. 326–343. ISSN: 0003-0554, 1537-5943. DOI: 10.1017/S0003055413000014.

[57] J. Knockel, M. Crete-Nishihata, J. Q. Ng, A. Senft, and J. R. Crandall. "Every Rose Has Its Thorn: Censorship and Surveillance on Social Video Platforms in China". In: *Proceedings of the 5th USENIX Workshop on Free and Open Communications on the Internet (FOCI 15)*. FOCI'15. Washington, D.C.: USENIX Association, Aug. 2015.

[58] J. Knockel and L. Ruan. "Measuring QQMail's automated email censorship in China". In: *Proceedings of the ACM SIGCOMM 2021 Workshop on Free and Open Communications on the Internet*. FOCI'21. Virtual Event USA: ACM, Aug. 27, 2021, pp. 8–15. ISBN: 978-1-4503-8640-1. DOI: 10.1145/3473604.3474560.

[59] J. F. Kurose and K. W. Ross. *Computer networking : a top-down approach*. Eighth edition, global edition. Publication Title: Computer networking : a top-down approach. Hoboken, N.J: Pearson, 2022. 648 - 649. ISBN: 978-1-292-40551-3.

[60] G. Lowe, P. Winters, and M. L. Marcus. *The Great DNS Wall of China*. Tech. rep. New York University, 2007. URL: https://censorbib.nymity.ch/pdf/Lowe2007a.pdf (visited on 05/04/2025).

[61] A. Master and C. Garman. "A Worldwide View of Nation-state Internet Censorship". In: *Proceedings of Free and Open Communications on the Internet 2023 (FOCI)*. Free and Open Communications on the Internet (FOCI'23). Vol. 2. Lausanne, Switzerland, July 10, 2023, pp. 1–21.

[62] A. A. Niaki, S. Cho, Z. Weinberg, N. P. Hoang, A. Razaghpanah, N. Christin, and P. Gill. "ICLab: A Global, Longitudinal Internet Censorship Measurement Platform". In: *2020 IEEE Symposium on Security and Privacy (SP)*. 2020 IEEE Symposium

on Security and Privacy (SP). ISSN: 2375-1207. May 2020, pp. 135–151. DOI: 10.1109/SP40000.2020.00014.

[63] N. Niere, S. Hebrok, J. Somorovsky, and R. Merget. "Poster: Circumventing the GFW with TLS Record Fragmentation". In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. CCS '23: ACM SIGSAC Conference on Computer and Communications Security. Copenhagen Denmark: ACM, Nov. 15, 2023, pp. 3528–3530. ISBN: 9798400700507. DOI: 10.1145/3576915.3624372.

[64] *OONI: Open Observatory of Network Interference*. URL: https://ooni.org/ (visited on 03/28/2025).

[65] J. C. Park and J. R. Crandall. "Empirical Study of a National-Scale Distributed Intrusion Detection System: Backbone-Level Filtering of HTML Responses in China". In: *2010 IEEE 30th International Conference on Distributed Computing Systems*. 2010, pp. 315–326. DOI: 10.1109/ICDCS.2010.46.

[66] P. Pearce, B. Jones, F. Li, R. Ensafi, N. Feamster, N. Weaver, and V. Paxson. "Global Measurement of DNS Manipulation". In: *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, Aug. 2017, pp. 307–323. ISBN: 978-1-931971-40-9.

[67] phobos. *Torproject.org Blocked by GFW in China: Sooner or Later? | Tor Project*. June 21, 2008. URL: https://blog.torproject.org/torprojectorg-blocked-gfw-china-sooner-or-later/ (visited on 03/28/2025).

[68] V. Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczynski, and W. Joosen. "Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation". In: *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS)*. 26th Annual Network and Distributed System Security Symposium (NDSS). Jan. 2019. DOI: 10.14722/ndss.2019.23386.

[69] R. S. Raman, M. Wang, J. Dalek, J. Mayer, and R. Ensafi. "Network measurement methods for locating and examining censorship devices". In: *Proceedings of the 18th International Conference on emerging Networking EXperiments and Technologies*. CoNEXT '22. New York, NY, USA: Association for Computing Machinery, Nov. 30, 2022, pp. 18–34. ISBN: 978-1-4503-9508-3. DOI: 10.1145/3555050.3569133.

[70] R. Rambert, Z. Weinberg, D. Barradas, and N. Christin. "Chinese Wall or Swiss Cheese? Keyword Filtering in the Great Firewall of China". In: *Proceedings of the Web Conference 2021*. WWW '21. Ljubljana, Slovenia: Association for Computing Machinery, 2021, pp. 472–483. ISBN: 978-1-4503-8312-7. DOI: 10.1145/3442381.3450076.

[71] E. Rescorla. *HTTP Over TLS*. Request for Comments RFC 2818. Num Pages: 7. Internet Engineering Task Force, May 2000. DOI: 10.17487/RFC2818.

[72] E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. Request for Comments RFC 8446. Num Pages: 160. Internet Engineering Task Force, Aug. 2018. DOI: 10.17487/RFC8446.

[73] E. Rescorla, K. Oku, N. Sullivan, and C. A. Wood. *TLS Encrypted Client Hello*. Internet Draft draft-ietf-tls-esni-24. Num Pages: 53. Internet Engineering Task Force, Mar. 20, 2025. URL: https://datatracker.ietf.org/doc/draft-ietf-tls-esni (visited on 03/23/2025).

[74] *RIPE Atlas - RIPE Network Coordination Centre*. URL: https://atlas.ripe.net/ (visited on 04/27/2025).

[75] A. Segal. *China's Vision for Cyber Sovereignty and the Global Governance of Cyberspace*. NBR Special Report #87. The National Bureau of Asian Research, Aug. 25, 2020, pp. 85–100.

[76] *Shadowsocks | A fast tunnel proxy that helps you bypass firewalls.* URL: https://shadowsocks.org/ (visited on 03/21/2025).

[77] A. Shahbaz, A. Funk, J. Brody, K. Vesteinsson, G. Baker, C. Grothe, M. Barak, M. Masinsin, R. Modi, and E. Stutterlin. *Freedom on the Net 2023, The Repressive Power of Artificial Intelligence*. Freedom House. Nov. 2023. URL: https://freedomhouse.org/report/freedom-net/2023/repressive-power-artificial-intelligence (visited on 05/04/2025).

[78] *The Citizen Lab*. URL: https://citizenlab.ca/ (visited on 03/28/2025).

[79] *The Tor Project | Privacy & Freedom Online*. URL: https://torproject.org (visited on 03/21/2025).

[80] *Tor partially blocked in China | Tor Project*. URL: https://blog.torproject.org/tor-partially-blocked-china/ (visited on 05/02/2025).

[81] *United States and Canada | OpenNet Initiative*. 2010. URL: https://opennet.net/research/regions/united-states-and-canada (visited on 03/21/2025).

[82] B. VanderSloot, A. McDonald, W. Scott, J. A. Halderman, and R. Ensafi. “Quack: Scalable Remote Measurement of Application-Layer Censorship”. In: *27th USENIX Security Symposium*. 27th USENIX Security Symposium (USENIX Security 18). 2018, pp. 187–202. ISBN: 978-1-939133-04-5.

[83] V. Ververis, T. Ermakova, M. Isaakidis, S. Basso, B. Fabian, and S. Milan. “Understanding Internet Censorship in Europe: The Case of Spain”. In: *Proceedings of the 13th ACM Web Science Conference 2021*. WebSci ’21. New York, NY, USA: Association for Computing Machinery, June 22, 2021, pp. 319–328. ISBN: 978-1-4503-8330-1. DOI: 10.1145/3447535.3462638.

[84] Z. Wang, Y. Cao, Z. Qian, C. Song, and S. V. Krishnamurthy. “Your state is not mine: a closer look at evading stateful internet censorship”. In: *Proceedings of the 2017 Internet Measurement Conference*. IMC ’17: Internet Measurement Conference. London United Kingdom: ACM, Nov. 2017, pp. 114–127. ISBN: 978-1-4503-5118-8. DOI: 10.1145/3131365.3131374.

[85] *Web Filter Lookup*. FortiGuard Labs. URL: https://fortiguard.fortinet.com/webfilter (visited on 04/04/2025).

[86] *WebRTC*. WebRTC. URL: https://webrtc.org/ (visited on 05/02/2025).

[87] *What is encrypted SNI? | How ESNI works*. URL: https://www.cloudflare.com/learning/ssl/what-is-encrypted-sni/ (visited on 03/23/2025).

[88] T. Wilde. *Tor Bug 4185 Testing and Report*. Gist. Dec. 6, 2011. URL: https://gist.github.com/twilde/da3c7a9af01d74cd7de7 (visited on 03/28/2025).

[89] P. Winter and S. Lindskog. “How the Great Firewall of China is Blocking Tor”. In: *Proceedings of the 2nd USENIX Workshop on Free and Open Communications on the Internet (FOCI 12)*. FOCI’12. Bellevue, WA: USENIX Association, Aug. 2012.

[90] P. Winter, T. Pulls, and J. Fuss. “ScrambleSuit: a polymorphic network protocol to circumvent censorship”. In: *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*. CCS’13: 2013 ACM SIGSAC Conference on Computer and Communications Security. Berlin Germany: ACM, Nov. 4, 2013, pp. 213–224. ISBN: 978-1-4503-2485-4. DOI: 10.1145/2517840.2517856.

[91] M. Wu, J. Sippe, D. Sivakumar, J. Burg, P. Anderson, X. Wang, K. Bock, A. Houmansadr, D. Levin, and E. Wustrow. “How the Great Firewall of China Detects and Blocks Fully Encrypted Traffic”. In: *32nd USENIX Security Symposium*

(USENIX Security 23). Anaheim, CA: USENIX Association, Aug. 2023, pp. 2653–2670. ISBN: 978-1-939133-37-3.

[92] D. Xue, R. Ramesh, V. S. S, L. Evdokimov, A. Viktorov, A. Jain, E. Wustrow, S. Basso, and R. Ensafi. "Throttling Twitter: an emerging censorship technique in Russia". In: *Proceedings of the 21st ACM Internet Measurement Conference*. IMC '21: ACM Internet Measurement Conference. IMC '21. Virtual Event: Association for Computing Machinery, Nov. 2021, pp. 435–443. ISBN: 9781450391290. DOI: 10.1145/3487552.3487858.

[93] T. Zhu, A. Pridgen, and D. S. Wallach. "The Velocity of Censorship: High-Fidelity Detection of Microblog Post Deletions". In: *Proceedings of the 22nd USENIX Security Symposioum*. 22nd USENIX Security Symposioum. Washington D.C., USA, Aug. 14, 2013.

[94] J. Zittrain and B. Edelman. "Internet filtering in China". In: *IEEE Internet Computing* 7.2 (Mar. 2003), pp. 70–77. ISSN: 1941-0131. DOI: 10.1109/MIC.2003.1189191.

[95] J. L. Zittrain and J. G. J. Palfrey. *Internet Filtering in China in 2004 - 2005: A Country Study*. OpenNet Initiative. Apr. 2005. URL: https://opennet.net/sites/opennet.net/files/ONI_China_Country_Study.pdf (visited on 05/04/2025).