WinTOR is Coming

Adding application-layer feature randomization and traffic control to Tor's Snowflake pluggable transport to strengthen censorship resistance against the Great Firewall of China

Hayden Edelson
Information Security and Privacy
Research Paper

Abstract

The internet has become the single most important conduit for political dissent, individual expression, and information exchange in the world. While it is a potent tool for freedom-seekers and political activists, censorship around the world has intensified to limit its use. The profoundly negative ramifications of this posture toward internet freedom are evident in today's society—particularly in light of the COVID-19 crisis, one that has been exacerbated by government censorship and the suppression of free speech. Censorship circumvention tools, such as Tor, have emerged to protect individuals' access to the free and open internet, but sophisticated censorship regimes, such as China, have exerted tremendous effort to restrict access to it. One censorship circumvention tool under development is called Snowflake. Snowflake is a WebRTC-based pluggable transport designed to circumvent censorship through protocol obfuscation and collateral freedom. This paper proposes improvements to Snowflake that could enhance its censorship resistance without negatively impacting network speed or user anonymity.

1 Introduction

The Tor Project ("Tor ") is an organization devoted to internet anonymity and censorship circumvention. It is responsible for the maintenance of a server network, transport protocol, and browser application that serve to protect internet users' digital freedom. Most countries allow free and indiscriminate use of Tor (so long as that use is toward legal ends); however, some countries, such as China, Syria, and Iran, attempt to block Tor's use. The cohort of countries that block Tor varies widely in terms of resources and technological sophistication, but all represent oppressive—and in some cases, brutal—regimes that seek to suppress political activism and enforce strict media censorship.

China has the most sophisticated internet censorship apparatus in the world. Through the use of artificial intelligence [3], the cooperation of Chinese internet and telecom companies [13], and an extraordinary amount of manpower [12], the Chinese government has the capacity to inspect every single packet of data on the Chinese internet, amounting to near total surveillance of its 1.4 billion citizens' online activities. China's censorship program is known as the "Great Firewall of China" (GFW) [1]. One of Tor's newest weapons in the arms race against government censors is the Snowflake pluggable transport, a WebRTC-based protocol obfuscation module that attempts to make Tor traffic look like unremarkable WebRTC traffic. This research paper proposes technical enhancements to Snowflake that may enable it to better circumvent the GFW. The proposed enhancements include implementing protocol feature randomization and traffic control functionality in the application layer of the Tor browser. These improvements will help prevent packet fingerprinting and traffic analysis attacks while securing anonymity and digital freedom for all who wish to access the internet.

The remainder of the paper is structured as follows: an overview of Tor, an account of the conflict between Tor and internet censors and the technologies that resulted, and a discussion of the present proposal, concluding with ideas for future research.

1.1 Overview of the Tor Network

The Tor Network is a network of servers that supports an internet protocol and browser application, which allow users to maintain anonymity on the internet. The servers are run by volunteers all over the world. The Tor protocol works by routing traffic through multiple severs (also called "nodes") using secure connections and multiple layers of encryption [2]. Using this protocol, it is almost impossible for any malicious actors to track a user's activities on the network [2]. Figure 1 and the explanation below provide additional detail on how Tor works:

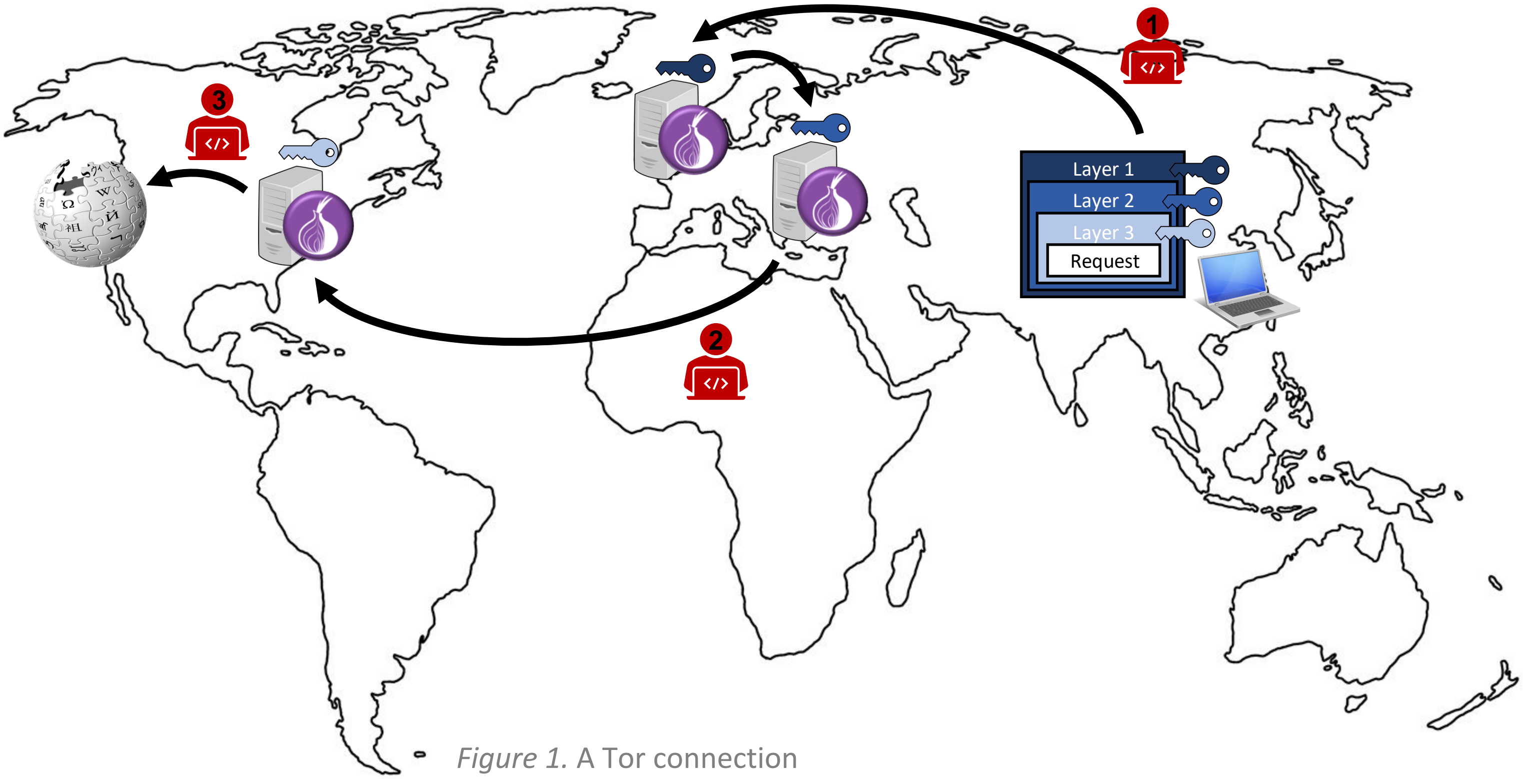


Figure 1. A Tor connection

When a client initiates a request over the Tor Network, it randomly selects three nodes through which to route its traffic. It then generates a set of encryption keys with each node and encrypts its request with three layers of encryption. The client then sends its message to the first node.

1. Upon receipt, the first node strips off the first layer of encryption, uncovering instructions to forward the message to the second node.
2. The second node strips off the second layer of encryption, uncovering instructions to forward the message to the third node.
3. The third node strips off the final layer of encryption and sees the message's final destination.
4. The destination server's response is sent back over the network and is re-encrypted at each node.

Importantly, no node in the network is privy to both the origin of the message and its final destination. Each individual node knows only (1) where the packet came from and (2) what its next stop is. Therefore, de-anonymized traffic surveillance proves extremely difficult. The following attack scenarios illustrate this fact:

- Attacker 1 can see that the client is communicating with a Tor node but cannot see the contents of the message nor its final destination.
- Attacker 2 can see only that there is some traffic going from one Tor node to another Tor node. He knows not the sender, recipient, nor content of the message.
- Attacker 3 can see that someone on the Tor Network is communicating with Wikipedia but does not know who that person is. This is how Tor protects its users' anonymity [2].

2 Background and Related Research

2.1 Tor and the Censorship Arms Race [3]

The "Censorship Arms Race" describes the back-and-forth, retaliatory conflict between Tor and censoring governments. It has gone through a number of different phases since it began around 2006, but the pace of escalation varies widely by government. The more motivated and sophisticated censors, such as China and Iran, are much more powerful adversaries than others, such as Turkey and Syria. Roger Dingledine, the founder of the Tor Project, describes it in three phases [3]:

- Phase 1: DPI, bridges, and pluggable transports;
- Phase 2: active probing, obfsproxy, and domain fronting; and
- Phase 3: of snowflake, obfs4, and decoy routing

2.1.1 Phase 1. Originally, Tor nodes were all publicly listed on servers called "Directory Authorities." When a client sought to bootstrap a Tor connection, it contacted a Directory

Authority, randomly selected three IP addresses, and established its network [2]. The public listing of IP addresses is necessary for the Tor network: nodes are volunteer-run, which means the network is often changing. Existing volunteers shut their nodes down, and new volunteers contribute to the network's expansion, so the centralized, public listing of node addresses is the most efficient way to make network information available to clients. Nodes listed in these public directories are called "relays" [2].

However, some governments soon realized that they could block the network by simply blacklisting these IP addresses. In response, Tor built up a network of nodes that it kept private. It distributed the IP addresses of these nodes, called "bridges," on a slow and selective basis, hoping simply that it would take a long time and a substantial investment for a government to block all of them [3]. Eventually, some did. However, the bridge network continues to grow and remains an important component of the Tor network today.

In addition, this phase of the Censorship Arms Race is marked by governments' use of deep packet inspection (DPI) to identify Tor packets on their local networks. DPI allows network sniffers to inspect the contents of data packets. While Tor packets are encrypted, so network sniffers can't see what's inside, DPI allows them to analyze packets' appearance and statistical qualities, such packet sizes and protocols. Based on that analysis, the censor can attempt to deduce the likelihood that a given packet is Tor, and can choose whether or not to filter it on that basis. As a result, Tor developed network protocol obfuscation modules called pluggable transports (PTs). PTs transmute the appearance of Tor packets to make them resemble other network protocols or otherwise network randomness, making them difficult for DPI to identify reliably [4].

2.1.2 Phase 2. Phase 2 of the Censorship arms race began in 2012 when China pioneered a censorship strategy known as "active probing" [3]. Active probing works in conjunction with DPI and IP blacklisting to identify Tor servers and block their IP addresses. In China, DPI is used to identify Tor packets and determine their destination servers' IP addresses. China's active probing system then initiates a connection with the server using the Tor protocol. If the server responds also using the Tor protocol, thus revealing itself to be a Tor server, China immediately blacklists its IP address.

In response, Tor expanded its arsenal of pluggable transports. The effort has yielded some success, but active probing remains an extremely powerful censorship mechanism. Tor's meek pluggable transport, described in section 2.3, is the only pluggable transport that works effectively in China today.

2.1.3 Phase 3. The current phase of the censorship arms race is in its very early stages. Obfs4 and Snowflake, both pluggable transports, will be described in section 2.3. Decoy routing attempts to bypass firewalls through covert proxy servers placed inside of network routers [25]. All three of these technologies are still in active development stages.

2.2 Censorship in China

China has the most sophisticated internet censorship regime in the world. According to the Council on Foreign Relations, Beijing maintains tight control over all internet and media content, and strictly enforces censorship rules through a number of government agencies and corporations [12]. Chinese internet companies are required to sign the “Public Pledge on Self-Regulation and Professional Ethics for China Internet Industry,” a pledge to serve the interests of the Chinese government. The pledge includes, among other provisions, “Refraining from producing, posting, or disseminating pernicious information that may jeopardize state security and disrupt social stability, contravene laws and regulations and spread superstition and obscenity. Monitor[ing] the information publicized by users on websites according to law and remov[ing] the harmful information promptly [13].”

China’s internet censorship apparatus is a backbone-level firewall known as the “Great Firewall of China” (GFW) [1]. The GFW employs a variety of censorship technologies, including artificial intelligence, IP blocking, deep packet inspection (DPI), and active probing [3]. The fully integrated suite of censorship technologies operates approximately as follows: an automated, AI-driven system conducts DPI on every packet of data on the Chinese internet. When it suspects a packet to be Tor, a GFW server pings the packet’s destination server using the Tor protocol. If the destination server responds using the Tor protocol, then the GFW immediately blacklists its IP address. Today, Tor servers do not respond unless the communicating server can provide some secret that only the Tor client would know. By not revealing themselves to be Tor servers, they can avoid being blocked. However, they are still forced to drop the connection, so active probing is still effective in preventing communications on the Tor network. Tor has had great difficulty developing new ways to bypass active probing [3].

3 Censorship Circumvention Technologies

3.1 Pluggable Transports

The below paragraphs offer brief summaries of Tor’s pluggable transports. There are currently two in operation—obfs4 and meek—and several under development, including snowflake [4]. Snowflake is the primary focus of this paper.

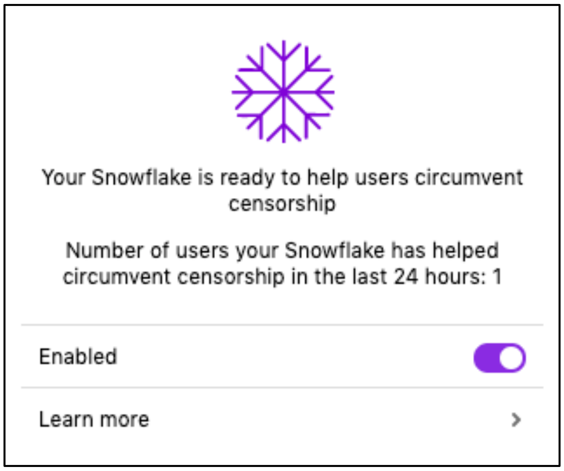
3.1.1 Obfs 4. Obfs4 works by obfuscating (hence the name) the Tor protocol to make Tor traffic look like random bytes. It also employs cryptographic security mechanisms to resist DPI and man-in-the-middle attacks, as well as the secret exchange defense against active probing [5].

Obfs4 works by forcing censors to accept high false positive filtering rates. When a censor encounters an obfs4 packet, its general reaction to the packet is “I don’t recognize this type of traffic" [3]. Therefore, a censor has to decide whether or not to filter all the packets that it can't neatly categorize. If it chooses to do so, it will inevitably filter a large number of packets that it should have let through; each incorrectly filtered packet represents some economic cost to the censor—in terms of productivity, access to information, or social unrest [17].

3.1.2 meek. meek utilizes a practice called "domain fronting" to make Tor traffic look like regular HTTPS traffic [6]. Domain fronting involves piping data through an actual HTTPS web server, such as AWS or Azure, such that the only way for censors to block these packets is by blocking the servers themselves [6][7].

The basis for domain fronting is a concept known as collateral freedom [7]. Collateral freedom is achieved by leveraging societally important technologies or services in such a way that internet censorship becomes economically prohibitive. For example, blocking major cloud service providers would alienate Chinese industry from a significant portion of the global information economy. However, domain fronting relies on the cooperation (or at least, the begrudging acquiescence) of the cloud service providers themselves, who have largely disallowed the practice. Azure is the only major cloud platform on which domain fronting still works, but Microsoft is expected to ban the practice at some point in the near future [8][9].

3.1.3 Snowflake. Snowflake is a WebRTC-based, peer-to-peer proxying system that allows internet users in free countries to act as proxy servers for Tor clients in highly censored countries [10]. The ultimate objective of Snowflake is to create a highly ephemeral and difficult-to-block network of servers. Dunna, O'Brien, and Gill found that the GFW tends to block Tor bridges for 12 hours after detection. After 12 hours, the GFW re-scans the bridges and unblocks them if they are no longer acting as Tor servers [1]. If enough users act as snowflakes, Tor builds up a cache of proxy servers that could enable it to outpace the Chinese government's IP blocking capabilities. In addition, Snowflake attempts to invoke collateral freedom, such that the only way to block Snowflake is to block WebRTC altogether or to block a sizable portion of global internet users [11].



Your Snowflake is ready to help users circumvent censorship

Number of users your Snowflake has helped circumvent censorship in the last 24 hours: 1

Enabled

Learn more

Figure 2. Snowflake browser extension

Figure 3 illustrates a Snowflake connection. The client communicates with a Broker, a server hosted on a third-party web service, via domain fronting. This interaction with the Broker is called the rendezvous [10]. The Broker then facilitates a connection to a Snowflake proxy, and the client communicates with the proxy using WebRTC [10]. A reasonable question is: if Snowflake relies on domain fronting, why not just use meek? The answer is, domain fronting costs money, so Snowflake is a much more cost-effective approach. Moreover, Snowflake is less reliant on domain fronting. Were domain fronting to become unavailable, meek would cease to function, whereas the rendezvous could likely be facilitated using steganography, a SOCKS proxy, or some other censorship circumvention technology.

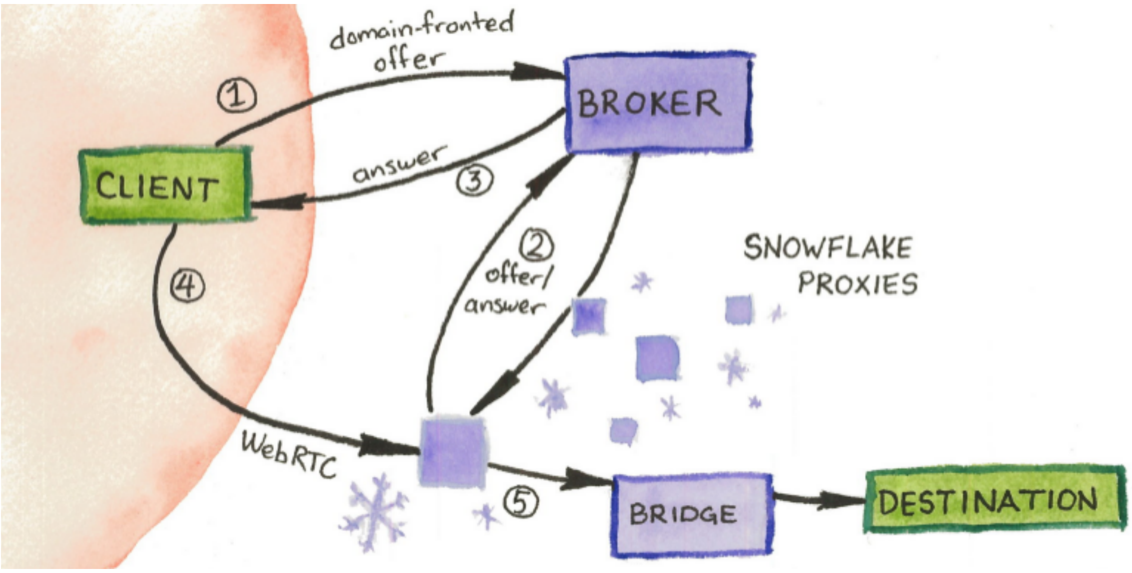


Figure 3. A Snowflake connection

3.2 Other Censorship Circumvention Technologies

3.2.1 Shadowsocks. A popular censorship circumvention tool in China is Shadowsocks. Shadowsocks is effectively a SOCKS proxy with high customizability, including server IP address and cipher suite customizability [14]. Shadowsocks is effective because every connection is different: each client is responsible for selecting its own proxy server and encryption algorithm, so every connection looks like a distinct HTTPS connection. If one server gets blocked, users can easily change their server IP address.

While Shadowsocks is an effective censorship circumvention tool, it has a number of drawbacks. First, it requires some technical sophistication: it requires an ability to establish a proxy server and a basic understanding of cryptographic algorithms. Proxy servers cost money, so it also requires an ability to pay. In addition, because of the uniqueness of each connection, and the fact that each connection only routes through one proxy server, Shadowsocks is not anonymizing. Nonetheless, it represents a valuable case study for Tor as an effective censorship circumvention technology.

3.2.2 Freenet and I2P. Tor is by far the most popular anonymity network in the world, but two other well-known anonymity networks are Freenet and the Invisible Internet Project (I2P). Both Freenet and I2P function as peer-to-peer anonymity networks. I2P employs a network routing paradigm known as garlic routing. Garlic routing is a variant of onion routing in which multiple messages are encrypted and sent together [16]. This method provides additional protection against packet timing and traffic analysis attacks [16]. Freenet is a peer-to-peer filesharing and website hosting network wherein users allocate a portion of their local disk space to file storage [15]. Collectively, this disk space is known as the Freenet Datastore. Freenet uses a high-latency, small-world network routing algorithm [26].

Freenet and I2P are similar in that they are both distributed, decentralized anonymity networks—unlike Tor, which is centralized via its directory authorities—and insulated darknets, meaning that neither provides Clearnet browsing capability—again, unlike Tor [15][16]. Both of these networks are considerably smaller than Tor, and are therefore subject to less scrutiny by censors. While potentially useful as examples of P2P networking services, Freenet and I2P generally respond to different demands than Tor does, and are therefore are not strong case studies in Clearnet censorship circumvention.

3.3 Real Quick: QUIC

QUIC is a relatively new network protocol designed as a potential successor to TCP. It was originally developed by Google and is now under the supervision of an IETF working group. The primary benefits of QUIC are that it offers lower latency and greater bandwidth efficiency than TCP [18], as shown in Figure 6. QUIC is implemented atop UDP, in the application layer. Some of the most significant features of QUIC are its multiplexing capability and its congestion control algorithm, which allow it to have remarkably high throughput compared to TCP [18]. Multiplexing is the act of combining multiple data streams into a single stream, and QUIC's congestion control algorithm helps resolve a significant connection latency-issue known as ACK ambiguity [18].

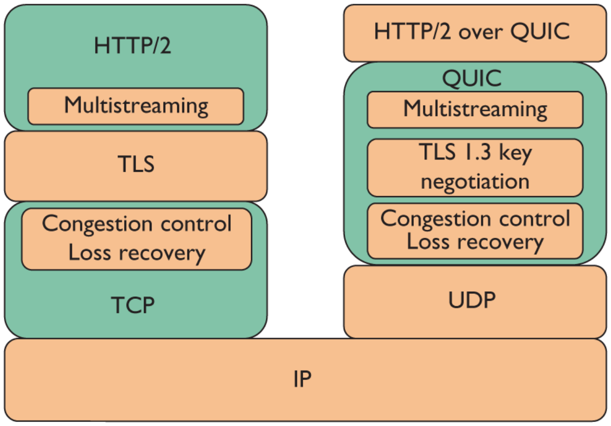


Figure 5. TCP vs QUIC

Similar to QUIC, WebRTC has a fairly "thick" layering because it provides a large amount of functionality. It can be implemented on top of TCP or UDP and has multiple tranches across the session and application layers. QUIC can be used as a model for how to implement network- and transport-layer functionality in the application layer.

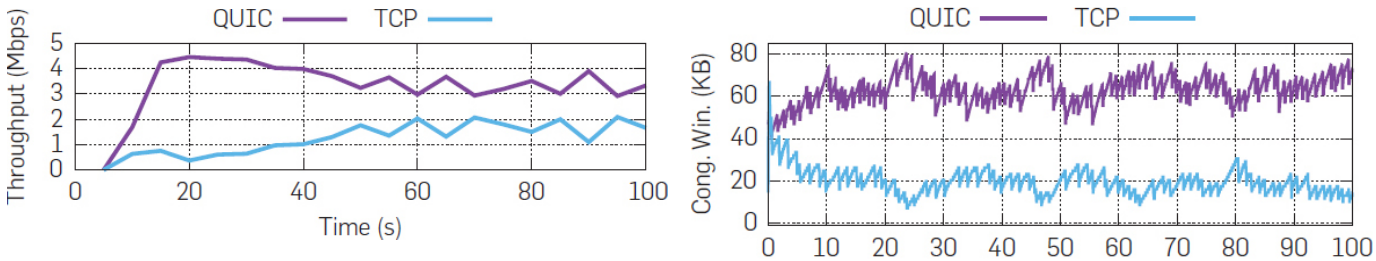


Figure 6. QUIC vs. TCP performance measurements

4 Hypothesis

As previously mentioned, Snowflake attempts to accomplish two very important anti-censorship tasks: (1) it attempts to create a large and highly ephemeral network of proxy servers, and (2) it attempts to invoke collateral freedom. The final hurdle to the successful deployment of Snowflake is minimizing its fingerprintability. David Fifield and Mia Epner found that many of the fingerprintable features of WebRTC connections exist in the Datagram Transport Layer Security (DTLS) layer [17]. These features include the DTLS version number, the ordered list of cipher suites and extensions offered by the client, the cipher suite chosen by the server, and the server's extensions [17]. Assigning to all Snowflake connections a fixed set of DTLS layer characteristics will inevitably lend itself to fingerprinting. However, attempting to randomize these characteristics in real-time, at the transport level, would significantly impact network speed and performance. Therefore, Tor can implement randomization functionality, including of cipher suites, packet sizes, and flow control, at the application level, allowing the browser to scramble the fingerprintable aspects of Snowflake packets without significant detriment to performance or users' anonymity.

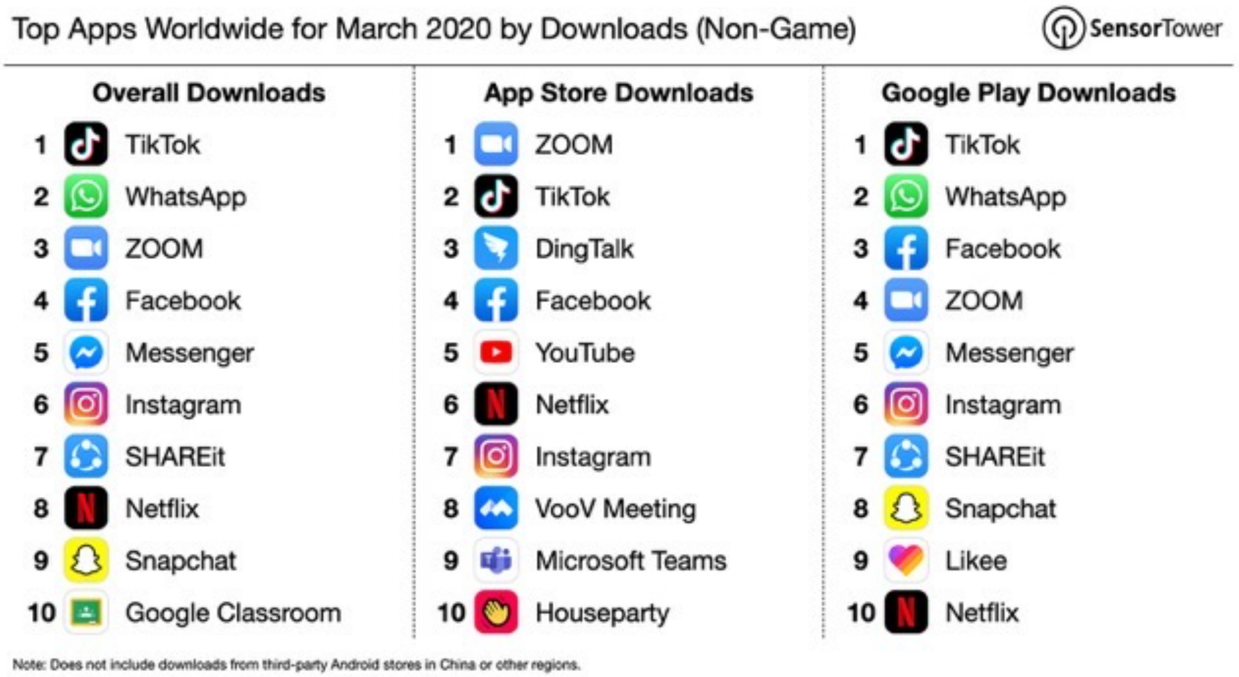
4.1 Application-layer Snowflake Randomization

By adding application-layer functionality, Snowflake could enable DTLS randomization on some periodic basis to reduce fingerprintability. For example, perhaps once per day or once per week, the client could randomize its list of cipher suites. Or, the client could randomly select a different flow control algorithm to make packet timing attacks less effective. All of this randomization could happen offline, while the user is not browsing the internet, so it has no effect on existing network connections. When the client connects to the internet, it interacts with the bridge as usual, using its randomized DTLS characteristics.

Adding randomization leverages collateral freedom. Censors are incentivized to fingerprint with as much specificity as possible [17]. If they don't, they risk inadvertently filtering packets that should have been allowed through the firewall. Each incorrectly filtered packet represents some economic cost to the censor—in terms of productivity, access to information, or social unrest [17]. Because China's firewall is largely AI-powered, and AI is particularly effective at recognizing patterns and repetitions, scrambling the protocol features as much as possible is likely to make it difficult for AI to detect reliably. Roger Dingledine, the founder of the Tor Project, has reported that in the past, changing a pluggable transport's list of cipher suites has actually resulted in a period of being able to bypass the GFW [3].

In the case of Snowflake, randomization will make most Snowflake packets on the network at any given time look a little different from one another. They still may not look exactly like any other WebRTC-based application's packets, but they will be difficult to fingerprint. Taking a lesson from Shadowsocks, differentiation from one packet to the next appears to support censorship circumvention.

While Shadowsocks has the distinct benefit of looking like HTTPS traffic, WebRTC may be approaching sufficient use levels to give the Chinese government some pause against heavy-handed filtering. In a post-COVID-19 world, WebRTC has the potential to become a particularly important technology. Throughout the COVID-19 pandemic, many workers in China have had to adjust to working from home [19]. As in the West, this process was facilitated by the use of videoconferencing services. In March 2020, many of the most downloaded apps in the world, as shown in Figure 7, were WebRTC-based. Zoom, Dingtalk, Microsoft Teams, and Houseparty all use some implementation of WebRTC [20]. In addition, Google's videoconferencing platform, Google Meet, is based on WebRTC, and Huawei uses WebRTC in its RCS messaging system [21][22]. Again, all of these platforms use WebRTC technology differently, so this hypothesis proposes a way to leverage this diversity and avoid fingerprinting.



Top Apps Worldwide for March 2020 by Downloads (Non-Game)

SensorTower

	Overall Downloads	App Store Downloads	Google Play Downloads
1	TikTok	ZOOM	TikTok
2	WhatsApp	TikTok	WhatsApp
3	ZOOM	DingTalk	Facebook
4	Facebook	Facebook	ZOOM
5	Messenger	YouTube	Messenger
6	Instagram	Netflix	Instagram
7	SHAREit	Instagram	SHAREit
8	Netflix	VooV Meeting	Snapchat
9	Snapchat	Microsoft Teams	Likee
10	Google Classroom	Houseparty	Netflix

Note: Does not include downloads from third-party Android stores in China or other regions.

Figure 7. Most downloaded apps, March 2020

5 Methodology

The following section describes an experiment to test the effectiveness of this hypothesis. The experiment involves a simple setup and a multi-day data collection period. The setup is followed by a description of important datapoints and data collection techniques and a discussion of the salient limitations and weaknesses of this proposal.

5.1 Setup

Set up a client machine on a virtual private server (VPS) in China and install the Tor browser. The Tor website is blocked in China, so the VPS can access the browser file via GitHub; a file transfer protocol, such as SFTP; or by fetching it from a website or other server that the Chinese government does not know about.

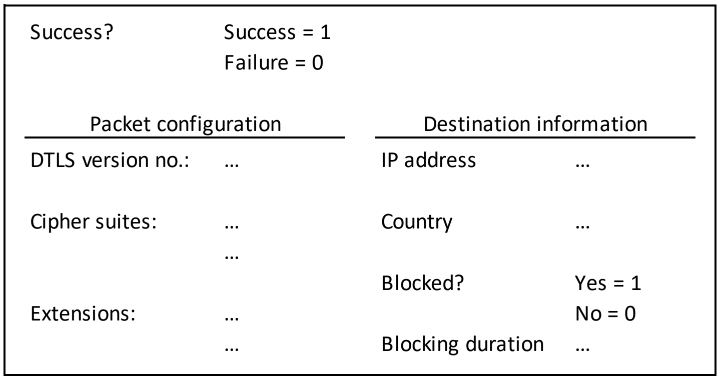
Enable the Snowflake browser extension on a number of different computers and host the Snowflake Javascript on a number of different websites around the world. For this experiment, Snowflake should be configured to allow the user to specify the proxy IP address. Attempt to bootstrap Snowflake connections using the test proxies.

5.2 Measurement

5.2.1 Test period. Over a period of several days or weeks, the VPS client should attempt to access the internet via the Tor browser, using Snowflake. The primary metric used in the evaluation of this hypothesis will be the ratio of successful Tor requests to total Tor requests. Therefore, the researchers should maintain a count of every successful and every unsuccessful connection.

In addition to the success rate, this study may produce valuable information related to China's censoring activities. Care should be taken to record the packet configurations of each request and the geographic locations and IP addresses of the Snowflake proxies for each connection attempt. Recording all of this information will allow Tor developers to determine if there are some locations or configurations that are more effective than others at circumventing the GFW.

Once a proxy's IP address appears to be blocked, the client should begin to measure the duration of the blocking. Dunna, O'Brien, and Gill observed that bridge servers tend to remain blocked for approximately 12 hours [1]. In the



Success?	Success = 1
	Failure = 0		

Packet configuration		Destination information	
DTLS version no.:	…	IP address	…
Cipher suites:	… …	Country	…

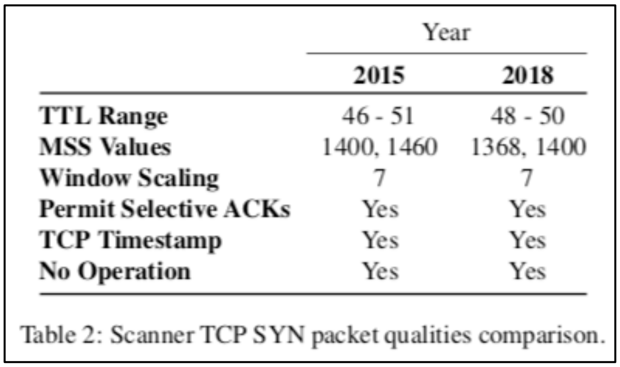
		Blocked?	Yes = 1
			No = 0

Extensions:	… …	Blocking duration	…

Figure 8. Record of connection attempt

context of the present experiment, certain questions arise: Would the duration be the same for a Snowflake proxy, which may just be an individual's computer? What do the active probing messages look like? What is the Snowflake proxy's response? The duration of the blacklisting can be measured by pinging the Snowflake IP address from the VPS every 60 seconds until a response is received. The Snowflake should be pinged on multiple ports to determine whether the entire IP address is blocked or only specific ports. This information should be recorded and compared across geographies. A sample connection record is shown in Figure 8.

As shown in Figure 9, Dunna, O'Brien, and Gill were able to establish a vague fingerprint for GFW scanners [1]. They observed that all of the GFW scanners they attracted were located in China. They determined that all of the scanners had similar packet qualities and TTL values, so they hypothesized that China maintains a network of proxies that forward packets from a centrally controlled system [1]. They observed that approximately 5% of SYN packets originated from IP address 111.202.242.93, and the other 95% were almost uniformly distributed, with no more than one or two packets from any one IP address [1]. The present study should attempt to follow up on these findings.



	Year	
	2015	**2018**
TTL Range	46 - 51	48 - 50
MSS Values	1400, 1460	1368, 1400
Window Scaling	7	7
Permit Selective ACKs	Yes	Yes
TCP Timestamp	Yes	Yes
No Operation	Yes	Yes

Table 2: Scanner TCP SYN packet qualities comparison.

Figure 9. GFW fingerprint

5.2.2 Full scale deployment. If the results of the test period are positive, the pluggable transport should be deployed at full-scale. An evaluation of this deployment would compare the average number of successful daily connections over some period of time before deployment to the average number of successful daily connections over some period of time after deployment (ideally, an equal period of time). Tor has a well-established methodology for counting network requests by country. Figure 10 shows estimated Tor usership in China over the last 12 months.

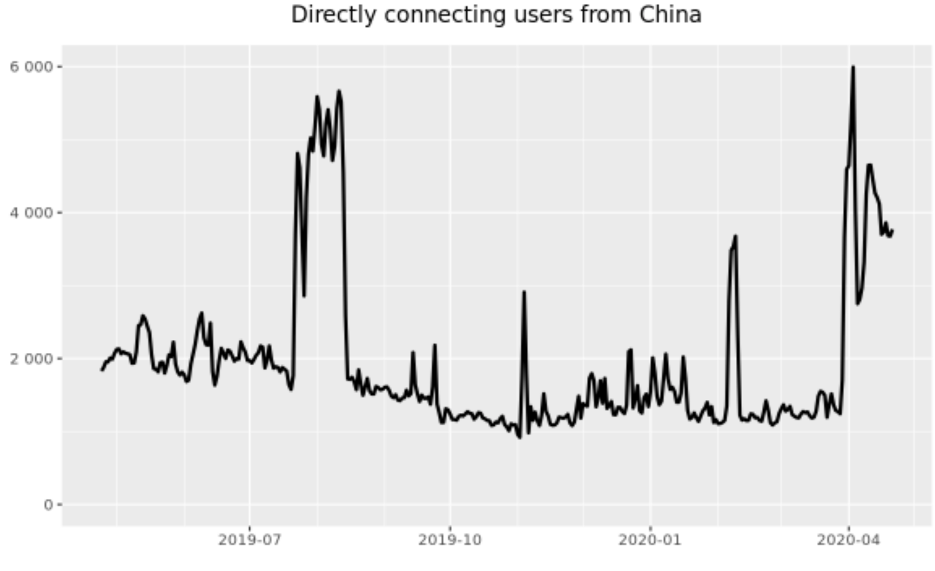


Figure 10. Estimated Tor usership in China

5.3 Limitations

5.3.1 Hosting on a VPS. As demonstrated by the “Public Pledge of Self-Regulation and Professional Ethics for China Internet Industry,” Chinese companies are responsible for assisting the government in internet censorship [13]. Therefore, it is possible that cloud service providers are independently responsible for filtering the traffic on their servers, and the filtering technology within these servers may be different than the filtering technology on China’s larger internet networks.

5.3.2 Location of the VPS. Different local networks in China may trigger different responses from the GFW. It is possible that the GFW monitors traffic in certain parts of the country more closely.

5.3.3 Location of the Snowflake servers. The GFW may respond differently to different geographic destinations. It is possible that the GFW filters certain destinations more than others. For this reason, it is important to establish Snowflake proxies in multiple countries, including China’s political allies (to the extent that Tor is available in those countries).

5.4 Weaknesses

There are certain attacks against which this recommendation would not be effective.

5.4.1 Banning WebRTC. If the Chinese government were to ban WebRTC in its entirety, Snowflake would cease to function in China. The current proposal hinges on the proposition that the Chinese government would not be willing to ban WebRTC due to the economic cost of doing so. This is likely a reasonable proposition, given the fact that major Chinese tech companies, such as Alibaba and Huawei, utilize WebRTC in important services.

5.4.2 Whitelisting. If the Chinese government were to whitelist specific WebRTC implementations, the current proposal would likely be ineffective. Enacting such a policy would have a significantly negative effect on the pace of innovation involving WebRTC and would likely lead to high collateral damage due to network randomness. However, this option does not seem entirely outside the realm of possibility for the Chinese government. If China did implement a policy of whitelisting, Tor could attempt to mimic one of the permissible implementations or could even attempt to utilize one of them as a steganographic medium for Snowflake.

5.4.3 Universal active probing. As of now, the GFW only attempts active probing if it suspects a packet to be Tor (or some other illicit connection). It is not clear what the response of a Snowflake proxy would be if probed by the GFW; however, if the GFW were able to identify Snowflake proxies, and were to actively probe the destination of every packet on the Chinese internet, this proposal would be likely ineffective. This scenario seems unlikely, however, given what the cost would likely be.

6 Future Research

Future research should explore the application of adversarial learning and generative adversarial networks to protocol obfuscation techniques [23]. Generative adversarial networks (GANs) are competitions between two models: (1) a generative model that analyzes a dataset and attempts to produce some output that looks like it came from the dataset, and (2) a discriminative model attempts to correctly classify the generative output as real or fake [24].

These learning techniques could be used to hone protocol obfuscation and traffic manipulation techniques that are intended to make Tor traffic look like some other, standard protocol traffic, such as HTTPS [23].

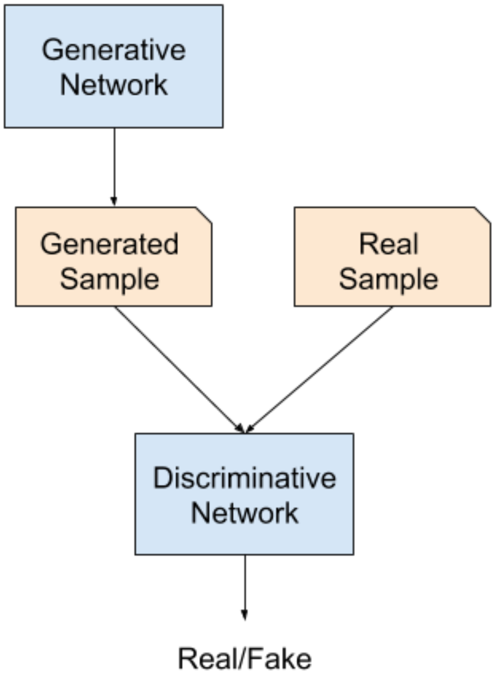


Figure 11. Generative adversarial networks

7 Conclusion

Globally, internet freedom is on the decline [27]. Authoritarian governments, such as China's, seek to suppress free speech and political activism—to enact invasive surveillance practices and cover up human rights violations. As tragically demonstrated by the COVID-19 crisis, humankind's global connectivity necessitates unmediated access to information and government transparency. This research paper proposes a series of feature enhancements to Tor's Snowflake pluggable transport to make it a more effective censorship circumvention tool, and to help guarantee access to the free and open internet for all.

Works Cited

[1] A. Dunna, C. O'Brien, and P. Gill, "Analyzing China's Blocking of Unpublished Tor Bridges," in *Workshop on Free and Open Communication on the Internet @ USENIX Security Symposium 2018, Baltimore, MD, USA, August 15-17, 2018.*

[2] "Tor Overview," *Tor Project.* [Online]. Available: https://2019.www.torproject.org/about/overview

[3] R. Dingledine, *The Tor Censorship Arms Race: The Next Chapter*, DEF CON 27, August 8-11, 2019, Las Vegas, NV.

[4] "Tor: Pluggable Transports," *Tor Project.* [Online]. Available: https://2019.www.torproject.org/docs/pluggable-transports.html.en

[5] "obfs4 Tranport Evaluation," *Tor Project.* [Online]. Available: https://trac.torproject.org/projects/tor/wiki/doc/PluggableTransports/Obfs4Evaluation

[6] "meek," *Tor Project.* [Online]. Available: https://trac.torproject.org/projects/tor/wiki/doc/meek

[7] D. Fifield, C. Lan, R. Hynes, P. Wegmann, and V. Paxson, "Blocking-resistant communication through domain fronting," in *Privacy Enhancing Technologies Symposium 2015, Philadelphia, PA, USA.* PoPETS 2015 pp. 1-19.

[8] C. MacCarthaigh, "Enhanced Domain Protections for Amazon CloudFront Requests," AWS Security Blog, April 27, 2018. [Online]. Available: https://aws.amazon.com/blogs/security/enhanced-domain-protections-for-amazon-cloudfront-requests/. [Accessed April 11, 2020].

[9] Steph, "Domain Fronting is Critical to the Open Web," Tor Blog, May 4, 2018. [Online] Available: https://blog.torproject.org/domain-fronting-critical-open-web. [Accessed April 13, 2020].

[10] "Tor: Snowflake," *Tor Project.* [Online]. Available: https://trac.torproject.org/projects/tor/wiki/doc/Snowflake

[11] "Snowflake Technical Overview." https://keroserene.net/snowflake/technical/

[12] B. Xu and E. Albert, "Media Censorship in China," *Council on Foreign Relations,* Feb. 17, 2017. [Online]. Available: https://www.cfr.org/backgrounder/media-censorship-china. [Accessed April 12, 2020].

[13] Public Pledge of Self-Regulation and Professional Ethics for China Internet Industry, 2002.

[14] S.D.T, "Shadowsocks: A secure SOCKS5 proxy," Jan. 4, 2019.

[15] I. Clarke, "A Distributed Decentralised Information Storage and Retrieval System," White Paper, Division of Informatics, Univ. of Edinburgh, Edinburgh, Scotland, 1999.

[16] "Introduction," *I2P.* [Online]. Available: https://geti2p.net/en/docs/how/tech-intro

[17] D. Fifield, M. Epner, "Fingerprintability of WebRTC," May 27, 2016.

[18] A. M. Kakhki, S. Jero, D. Choffnes, C. Nita-Rotaru, and A. Mislove, "Taking a Long Look at QUIC: An Approach for Rigorous Evaluation of Rapidly Evolving Transport Protocols," *Communications of the ACM,* Vol. 62, No. 7, pp. 86-94, Jul. 2019.

[19]L. H. Liang, "How Covid-19 led to a nationwide work-from-home experiment," *BBC*, Mar. 8, 2020. [Online]. Available: https://www.bbc.com/worklife/article/20200309-coronavirus-covid-19-advice-chinas-work-at-home-experiment

[20] "Why not meet via WebRTC?" *AV Magazine,* May 11, 2016. [Online]. Available: https://www.avinteractive.com/features/comment/why-not-meet-via-webrtc-11-05-2016/

[21] R. Lidstone. "Huawei Introduces WebRTC-Based Rich Communications Capability Exposure Gateway," *WebRTC World,* Apr. 25, 2013. [Online]. Available: http://www.webrtcworld.com/topics/webrtc-world/articles/335730-huawei-introduces-webrtc-based-rich-communications-capability-exposure.htm

[22] S. Y. Son, *WebRTC World Tour 2019,* W3C HTML5 Conference, Oct. 11, 2019, Seoul, South Korea.

[23] S. Sheffey and F. Aderholdt, "Improving Meek with Adversarial Techniques," in *Workshop on Free and Open Communication on the Internet @ USENIX Security Symposium 2019, Santa Clara, CA, USA, August 14-16, 2019.*

[24] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu. D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, "Generative Adversarial Nets," in *Proceedings of the International Conference on Neural Information Processing Systems, NIPS 2014, Montreal, CA, December 8-13, 2014.*

[25] J. Karlin, D. Ellard, A. W. Jackson, C. E. Jones, G. Lauer, D. P. Mankins, W. T. Strayer, "Decoy Routing: Toward Unblockable Internet Communication," in *Workshop on Free and Open Communication on the Internet @ FOCI 2011, San Francisco, CA, USA, August 8-12, 2011.*

[26] O. Sandberg, "Distributed Routing in Small-World Networks," in *Proceedings of the Workshop on Algorithm Engineering and Experiments, ALENEX 2006, Miami, FL, USA, January 21, 2006.*

[27] "Freedom on the Net 2019," Freedom House. November 4, 2019.