CENSORSHIP-CIRCUMVENTION TOOLS AND PLUGGABLE TRANSPORTS

Dan Bateyko*

CITE AS: 6 GEO. L. TECH. REV. 335 (2022)

TABLE OF CONTENTS

I. Introduction .................................................................................................................. 335
II. Tor Bridges .................................................................................................................... 337
III. Deep Packet Inspection .............................................................................................. 339
IV. The Pluggable Transports Model ................................................................................ 341
    A. Meek: A Diverting Strategy .................................................................................... 342
    B. Obfs4: A Randomizing Strategy ............................................................................. 343
V. Pluggable Transports in Practice: 2020 Internet Disruptions in Belarus .................. 345
VI. Conclusion .................................................................................................................... 349

I. INTRODUCTION

This Technology Explainer covers pluggable transports, 1 an obfuscation tool that disguises a user’s network traffic to avoid certain traffic analysis and detection methods.2 Pluggable transports are like plugins—the term refers to their ability to “plug in” and swap out, which affords developers

---


2 For developments to pluggable transports, see PLUGGABLE TRANSPORTS, supra note 1.
and users the chance to try different circumvention strategies in the same app once one method stops working. Software developers created pluggable transports in reaction to censors targeting and blocking anonymity tools used to circumvent censorship. In 2012, the Tor Project, a non-profit which maintains the anonymity software Tor, developed an early version of the pluggable transports concept to circumvent censorship of the Tor network in Iran and China. Since then, developers have built on the idea, some with support from U.S. government agencies like the U.S. Agency for Global Media and the Department of State, which funds projects to gain ground in the “arms race of protocol blocking.” This Explainer discusses pluggable transports featured in the Tor Browser, but many of the same principles apply to other software—anonymity tools, like Psiphon, messaging apps, like Signal and Telegram, and VPN providers, like Tunnelbear, have all made use of pluggable transports or similar obfuscating strategies.

---


8 Previous Technology Explainers on VPNs and the Tor Network provide helpful background. See, e.g., John Park, *VPN: Privacy and Anonymity for All*, 2 GEO. L. TECH. REV. 129 (2017); Kyle Swan, *Onion Routing and Tor*, 1 GEO. L. TECH. REV. 110 (2016).

This Explainer focuses on two pluggable transports: meek and obfs4. These two come built-in to the Tor Browser and are the most popular among Tor users. Part II briefly discusses the steps a Tor Browser user would take to connect to a blocked resource without using pluggable transports. Part III explains how censors use a method called deep packet inspection to analyze network traffic and block those vanilla Tor connections. Part IV describes the pluggable transports model, detailing in particular how meek and obfs4 allow users to disguise Tor connections and evade deep packet inspection-based surveillance and censorship. Part V provides a real-world example of how people experiencing network censorship use pluggable transports. Finally, in Part VI, the Explainer concludes with a brief speculation on the future of these obfuscation strategies.

II. TOR BRIDGES

To browse the web, watch online videos, or chat on a messenger app, a user typically needs to connect to a server. An “on-path attacker” puts themselves on the path between the user and the server to eavesdrop on the user’s communication. Internet service providers (ISPs) which manage the user’s network have a particularly privileged vantage point from which to observe the connection. This technology explainer considers examples of government network censorship, in which the state requires or compels ISPs


10 For an in-depth look into these technologies see Dingledine, supra note 6. See David Fifield et al., Tor’s Usability for Censorship Circumvention. ELECTR. ENG’G & COMPUT. SCIENCES UNIV. OF CALIFORNIA AT BERKELEY (May 12, 2016), https://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/ECECS-2016-58.pdf [https://perma.cc/8DUS-78CC]; Home Page, PLUGGABLE TRANSPORTS, https://www.pluggabletransports.info/ [https://perma.cc/G5V7-FRBT].


12 Throughout this Explainer, I choose to foreground peoples’ use of these tools by using the word “user” rather than “client,” which may be unfamiliar to some readers. For the same reason, I focus on the web browser experience in the Tor Browser, though an individual may connect to a service by other means.

to monitor and filter Internet connections, and censorship circumvention through Tor Browser.

When a user starts Tor Browser, the program retrieves a list of currently available Tor network servers, known as relays. The browser asks for a list from special servers called “directory authorities,” which are designed to maintain and publish a list of relays. These Tor relays are public. Consequently, a censor can find the list of relays and their IP addresses much the same as a user can. An ISP watching the Tor user’s traffic can then tell when the user requests to connect with one of these relays and could choose to block the user’s access. By finding relays and blocking traffic to-and-from them, the censor can block people from using the Tor network. In some cases, censors may even block the directory authorities themselves, disrupting a user’s ability to retrieve the list of relays in the first place.

Anticipating censors that block public relays, the Tor Project created “bridges.” Bridges are private relays; the Tor network does not distribute a list of these bridges in public view. Instead, lists of bridges are kept in a separate directory, called the bridge authority, and a separate distribution database called BridgeDB grants access to those bridges. To receive a bridge address from the BridgeDB and connect, users must overcome a hurdle: for example, solving a CAPTCHA in the Tor Browser, emailing the Tor Project directly to request one, or asking a trusted friend for an address. These obstacles are not

---

too burdensome, but they make the censor’s task of finding bridges to block harder.\textsuperscript{21}

\begin{figure}
\centering
\includegraphics[width=\textwidth]{bridge_interface.png}
\caption{Example of the user-interface enabling users to connect to a bridge.}
\end{figure}

But a censor can block more than just known relays. Even if a censor doesn’t know about a bridge before the user connects to it, the censor can inspect the user’s web traffic, determine that the person is using the Tor network, and cut them off.\textsuperscript{22} To do so, the censor uses deep packet inspection, a method covered in the next section.

\section{Deep Packet Inspection}


Internet networks send data in packets. A packet’s format starts with a “header,” which includes the IP addresses of the sender and recipient. The packet’s “payload” is its contents. Like an address written on the outside of a letter, a packet’s header can be read without actually examining its contents. An on-path attacker can see those headers, check the addresses against a block list, and drop the connection. This practice of observing traffic to spot the sender’s and recipient’s IP addresses is “shallow packet inspection.”

However, a censor observing the path between the sender and receiver might want more information than just the parties involved to determine if the connection merits blocking. For example, the censor may want to block all connections to a certain app, regardless of who is using it. In such cases, censors can go a step further and read other layers of the packet, including the payload. This kind of inspection goes deeper and is called “deep packet inspection.”

Deep packet inspection (DPI) is a process of inspecting packets to extract information and look for patterns in Internet traffic. DPI requires specialized hardware that sits on the path between the user and the server. Although ISPs use DPI routinely to manage traffic—for example, to identify security threats or change the rate of traffic to improve network efficiency, like throttling videos—ISPs can also use DPI to surveil and censor. One pair of scholars compared an ISP inspecting packets to a postal worker who opens letters, forwards illegal material to the police, and destroys any letters with prohibited content, all without the letter-writer’s knowledge.

Until the mid-2010s, web traffic between users and websites was largely unencrypted, carrying an increased risk of network connections exposing sensitive information. Only recently has the number of websites adopting an encrypted web protocol, HTTP over TLS (also known as HTTPS), surged—an improvement which mitigates DPI’s capability to read

---


24 ANNABEL Z. DODD, ESSENTIAL GUIDE TO TELECOMMUNICATIONS 26 (2019).


communications. Many apps, like Gmail and Drive, also protect against DPI by encrypting data transiting over the Internet. But an ISP using DPI need not break encryption to determine what traffic to block. In some cases, an ISP can identify when a person is using a certain app, like a VPN, by looking at other qualities of Internet traffic, like the protocol in use or patterns in the rate of traffic. An ISP could also choose to interfere with all HTTPS traffic on a network, giving users a Hobson’s choice—find a service with an unsecure, unencrypted connection to use or to not use the Internet at all. To identify Tor users, an ISP using DPI can distinguish vanilla Tor traffic by its particular traffic characteristics. Because recognizing Tor traffic is trivial to an ISP conducting DPI, the Tor Project introduced additional obfuscation strategies.

IV. THE PLUGGABLE TRANSPORTS MODEL

Developers have created several pluggable transports, each with a different strategy for dodging censorship. As one pluggable transport developer explains, these strategies generally fall into two categories: “The first is to mimic some content that the censor allows, like HTTP or email. The second is to randomize the content, making it dissimilar to anything that the censor specifically blocks.” Pluggable transports in the first category include those that camouflage traffic by adopting traffic characteristics of popular


30 “Even if you are using OpenVPN on the https port, and not using a well-known VPN host (which could be blocked by IP and/or DNS), DPI can inspect the traffic and identify it as OpenVPN, and block it based on that inspection.” PLUGGABLE TRANSPORTS, https://www.pluggabletransports.info/how/ [https://perma.cc/YDT9-CGBF]; Wiley & Oliver, supra note 3, at 3.


32 Matic et al., supra note 22.

apps, like Skype, or by routing traffic through popular cloud providers like Google or Amazon, which the censor chooses not to block; the second category include those that scramble network traffic to make it difficult for the censor to compare traffic to what it wants to block.

A. Meek: A Diverting Strategy

Released publicly on Tor Browser in August 2014, Meek makes traffic between a user and a blocked website look like something the censor allows.34 To do so, Meek hides identifiable, forbidden traffic like Tor connections in permissible HTTPS traffic.35 To prevent censors from seeing that a user is connected to a blocked service like a Tor bridge, Meek uses a domain-fronting strategy. “Domain fronting” is a technical strategy that turns cloud providers’ content delivery networks (CDNs) into go-betweens for the user and the blocked service. This strategy fools a censor into thinking a user is merely connected to a popular cloud provider like Amazon or Google. Meek creates a special HTTP over TLS request listing two recipient addresses: one to a “front domain” belonging to a cloud provider, which is placed in the server name indication field of the request, and one to the real intended address, encrypted in the payload and hidden from view.

The request first goes to a cloud provider’s CDN. The CDN receives the second, secret address and then forwards the message to that address, a server the cloud provider hosts called a reflector. That reflector then “reflects” or forwards the request again to a bridge, and the Tor user is connected to the network.36 Cloud providers like Amazon, Google, and Microsoft do not explicitly agree to act as such intermediaries but allow it incidentally due to quirks in how those companies route traffic. Domain fronting as a strategy relies on the censor’s unwillingness to block major cloud providers like Google out of fear of harming legitimate Google users or damaging market activity, a consequence referred to as “collateral damage.”37

---

Meek, once seen as a “gold standard,” now sees less use. In April 2018, Russian authorities blocked over a million Google and Amazon IP addresses in a brief attempt to enforce a nationwide block against Telegram, a messaging app using the domain-fronting strategy. While those disruptions were short-lived, soon after, Amazon and Google claimed to remove the technical features required for domain fronting to work, citing concerns unrelated to the Russian disruptions, like long-running abuse of domain fronting by malicious actors and routine technical changes to their networks. At the time of writing, Meek makes use of Microsoft’s Azure servers, which continue to allow domain fronting.

B. Obfs4: A Randomizing Strategy

Obfs4 is another pluggable transport built-in to the Tor Browser. Released in April 2015, the Tor Project estimates that obsf4 is the most-used pluggable transport. Obfs4 is a “look-like-nothing” protocol that attempts to make network traffic “unclassifiable” to a censor.

38 PLUGGABLE TRANSPORTS, supra note 1.
41 Circumvention, supra note 11.
As explained supra in Part IV, an ISP using DPI can determine patterns in traffic like Tor traffic and block the connection. But an ISP often limits its search for patterns to the first few packets sent across a connection because analyzing more at scale requires capturing that data in costly storage. Aware of this limitation and other faults in DPI, obfs4 works in several ways to thwart detection. The protocol randomizes packet lengths and packet arrival times—both of which censors associate with certain applications. obfs4 also adds a layer of encryption to make all traffic appear to be random data; a censor may permit such randomized traffic so long as it doesn’t match a blocked protocol.

However, a censor may become suspicious of traffic that appears to follow no patterns, especially if the traffic it normally sees on the network follows regular patterns. A censor might even wonder why, among seemingly pattern-less traffic, some connections seem to have a particularly unusual degree of randomness. In such cases, a censor, having found some traffic suspicious, might take unusual steps to rout out forbidden traffic. For instance, advanced actors pretend to be a Tor user and send connection requests to a suspicious server; if the censor can connect to the server using a blocked protocol, the censor knows to block the server. Obfs4 helps guard against this “active probing.” When a user requests an obfs4-supported bridge, the bridge’s address is given such that both the user and the server share a secret; an obfs4 server will only communicate with users that have proven knowledge of this shared secret and refuse other requests to connect.

46 Id. In 2016, Kazakhstan blocked Obfs4 traffic, possibly by making probabilistic guesses based on obfs4’s own time signatures. The protocol continues to be developed and refined. See David Fifield, Comment on “Allot Communications Blocking Vanilla Tor, Obfs4, and Meek in Kazakhstan”, TOR PROJECT (Nov. 14, 2016), https://gitlab.torproject.org/legacy/trac/-/issues/20348#note_2229553 [https://perma.cc/3MY9-PYOK] (cited in PLUGGABLE TRANSPORTS, supra note 1).
47 Yongzhong He et al., Detection of Tor Traffic Hiding Under Obfs4 Protocol Based on Two-Level Filtering, 2ND INT’L CONFERENCE ON DATA INTELLIGENCE AND SEC. 196 (2019).
49 Ensafi supra note 21.
50 For a more in-depth explainer on secret sharing and public key encryption, see generally Weisiyu Jiang, Public Key Encryption, 1 GEO. L. TECH. REV. 105 (2016).
V. PLUGGABLE TRANSPORTS IN PRACTICE: 2020 INTERNET DISRUPTIONS IN BELARUS

On the day of the 2020 Belarusian presidential election, authorities disrupted Internet access across the country.51 Citizens on their way to vote reported trouble connecting to the Internet, particularly to messaging apps like Telegram, WhatsApp and Viber.52 At 8:00 PM Minsk time, the polls closed.53 Soon after, as exit polling pointed to a landslide victory for the incumbent, President Alexander Lukashenko, the disruptions worsened.54 As riot police...


broke up crowds and detained protesters that had gathered to protest election tampering. Although representatives from Belarus’ publicly-owned telecom blamed cyberattacks for the Internet outages, independent watchdogs later reported evidence that authorities disrupted connectivity using a U.S. technology firm’s DPI technology. Human rights observers characterized these disruptions as an attempt to prevent protesters from coordinating demonstrations and documenting police brutality. The disruptions continued for days.

But people found workarounds. Capital residents used VPNs, although some users reported unreliable connections. Others installed censorship-circumvention software like Psiphon and the Tor Browser. As connections faltered, Minsk residents distributed USB drives with installation files, posted leaflets with QR codes linking to the circumvention software, and shared download links over Telegram whenever access permitted. Had the Belarusian government enforced its policies effectively, citizens would not have been able to use these censorship circumvention tools at all. Since 2015, the Ministry of Telecommunications has banned “proxy-servers and...
anonymous networks like Tor” and has been attempting to block Tor connections since 2016. Yet these anti-censorship tools worked and gained popularity fast. Two days after the election, Psiphon reported 1.76 million users in the country, purportedly 1 in every 3 Belarusians with Internet access. Protesters even managed to use Telegram, supposedly blocked; one survey found 45% of people in Telegram’s protest chats were still online during the disruptions on August 11. These users may have benefitted from Telegram’s own obfuscation software.

---


65 “We enabled our anti-censorship tools in Belarus so that Telegram remained available for most users there. However, the connection is still very unstable as Internet is at times shut off completely in the country.” Pavel Durov (@durov), TWITTER (Aug. 10, 2020, 3:56 PM), https://twitter.com/durov/status/1292912756233048064?lang=en [https://perma.cc/KJL5-7DGR].
Google saw a massive loss in traffic to its services, like YouTube, starting August 9, 2020.66

On August 9, the estimated number of Belarusians connecting directly to the Tor network dropped significantly.67

---

66 Google, supra note 56.
Figure 5: However, Belarusians could still access the Tor network by using pluggable transports built into the Tor Browser. The estimated number of Belarusians connecting via such tools jumped significantly during the block.68

VI. CONCLUSION

Censors and circumventors will both likely adopt new strategies as old ones fail. Already, programmers are developing machine learning and probability-based analyses to improve the speed and accuracy of detection methods, posing further challenges for obfuscation tool developers.69 Some nations may spur advances through investment in methods and technologies to spot obfuscating users, particularly as obfuscation tools become associated with criminality. For example, in one leaked document, The U.S. National Counterterrorism Center listed apps that use pluggable transports like the Tor Browser, Psiphon, and Tunnelbear as potential indicators an individual is engaged in “terrorist use of the dark web.”70 The agency suggests that obfuscation tools hide “dark-web software downloads and entry node list

70 NAT’L COUNTERTERRORISM CTR., NCTC 035601, POTENTIAL INDICATORS OF TERRORIST USE OF THE DARK WEB (2019) [https://perma.cs/DWC6-M6T7].
requests [that otherwise] might be detectable through a review of ISP logs"\(^{71}\) and that DPI could uncover terrorist suspects hiding their anonymous online activity through seemingly normal traffic.\(^{72}\)

In response, researchers continue to investigate ways to improve software and develop new pluggable transports. One such pluggable transport is Snowflake, which was recently released to Tor users.\(^{73}\) Developers for messaging apps seeking to protect uninterrupted access to users may integrate and build on censorship circumvention strategies.\(^{74}\) Lastly, widespread adoption of stronger standards like TLS version 1.3 on websites and in modern web browsers will further mitigate a censor’s ability to track and censor, possibly attenuating the use cases for censorship circumventing tools altogether.\(^{75}\)

\(^{71}\) Id.

\(^{72}\) NAT’L COUNTERTERRORISM CTR., supra note 70.

