MassBrowser:
Unblocking the Web for the Masses, By the Masses

Milad Nasr[*], Hadi Zolfaghari[*], and Amir Houmansadr
University of Massachusetts Amherst
{*milad,amir*}*@cs.umass.edu*
Project Website: https://massbrowser.cs.umass.edu/

Abstract

Existing censorship circumvention systems fail to offer reliable circumvention without sacrificing their users' QoS, or undertaking high costs of operation. We design a new circumvention system, called MassBrowser, with the objective of addressing such practical weaknesses of existing designs. Our system is based on a new design principle, called "the separation of properties," that states that circumvention systems should be tailored for circumvention as opposed to offering additional properties like anonymity. We combine various state-of-the-art circumvention techniques to make MassBrowser significantly resistant to blocking, while keeping its cost of operation small ($0.001 per censored client per month).

We have built and deployed MassBrowser as a fully operational system with end-user software for regular Internet users (currently in beta release mode). A key part of MassBrowser's design is using non-censored Internet users to run volunteer proxies to help censored users. We perform the first user study on the willingness of typical Internet users in helping circumvention operators. We have used the findings of our user study in the design of MassBrowser to encourage wide adoption by volunteers; particularly, our GUI software offers high transparency, control, and safety to the volunteers.

1 Introduction

The Internet plays a crucial role in today's social and political movements by facilitating the free circulation of speech, information, and ideas; democracy and human rights throughout the world critically depend on preserving and bolstering the Internet's openness. Consequently, repressive regimes, totalitarian governments, and corrupt corporations regulate, monitor, and restrict the access to the Internet, which is broadly known as Internet *censorship*. The techniques commonly used to enforce censorship include IP address blocking, DNS hijacking, and TCP content filtering [14, 38, 40, 60] to block access to certain destinations or to prevent certain forms of content from being transmitted. To ensure compliance and to detect undercover political/social activists, repressive regimes additionally utilize advanced networking tools, including deep packet inspection (DPI), to prevent the use of the censorship circumvention technologies by their citizens [35, 36, 57, 77].

To restore the openness of the Internet, researchers have designed and deployed an arsenal of tools [10, 14, 15, 31, 32, 41, 46, 50, 61, 72, 74, 80] that help users bypass censorship. Such tools, known as *circumvention systems*, deploy a variety of techniques ranging from IP indirection to onion routing to traffic obfuscation [38, 60].

Key shortcomings of existing systems: Unfortunately, existing circumvention systems suffer from one or all of the following weaknesses: *(1) Easily*

[*] The first two authors made equal contribution.

blocked: A majority of in-the-wild circumvention systems, including Tor, Lantern, Psiphon, and VPNs, work by setting up *proxy* servers outside the censorship regions, which relay traffic for censored users. Unfortunately, the proxies are implemented in a way that are easily blockable by the censors, e.g., due to using a small set of IP addresses that can get enumerated and blacklisted by the censors [57,59,75,77]. *(2) Costly to operate:* To resist proxy blocking by the censors, recent circumvention systems have started to deploy proxies on shared-IP platforms such as CDNs [44], App Engines [25], and Cloud Storage services [9], a technique broadly referred to as *domain fronting* [19]. This mechanism, however, is prohibitively expensive [45] to be used at large scale. *(3) Poor QoS:* Proxy-based circumvention systems like Tor and its variants [32, 43, 65] suffer from low quality of service (e.g., very high latencies and low bandwidths). This is primarily due to the imbalance between the bandwidth demand from censored users versus the bandwidth available by the proxies (e.g., Tor's ≈ 6500 relays need to proxy traffic for around two million daily users [58], while some users leverage Tor for bandwidth-extensive applications like BitTorrent. *(4) Hard to deploy:* Modern circumvention systems proposed in the academia are impractical to be used at large scale due to various reasons. For instance, decoy routing systems [31, 37, 80] require wide adoption by Internet ISPs, and tunneling systems [32, 34, 43, 65] can be disabled by third-party service providers they use for tunneling.

Our approach: In this paper, we present a new circumvention system that aims at addressing the shortcomings of existing circumvention solutions. We base our system on a *design principle* not employed by existing circumvention systems. Our principle, which we call the **separation of properties (SoP) principle**, states that *the key feature targeted by an effective circumvention system should be blocking resistance, and other features such as anonymity and browsing privacy should be left as optional to the users.* The SoP principle is based on the real-world observation [11, 20, 27, 66, 67] that the majority of censored users are solely interested in blocking resistance, e.g., to be able to access blocked news articles and be able to communicate through blocked social networks, but for the *majority* of the censored users properties like anonymity are not a concern. This is evident by the fact that "public" VPNs, "public" HTTP proxies, and centralized circumvention systems like Lantern [39] and Psiphon [53] are the most popular among censored users in China and Iran [66, 67] (when compared to privacy-preserving alternatives like Tor) despite the fact that they provide no anonymity or browsing privacy [27].

The SoP principle enables us to optimize the performance of a circumvention system around blocking resistance, and to offer features like anonymity and browsing privacy as options to the users (possibly by degrading the QoS). We will demonstrate how basing our design on SoP enables us to overcome the circumvention shortcomings discussed above. Note that while systems like VPNs and HTTP proxies do not aim for anonymity/privacy, they *do not leverage* the SoP principle in optimizing censorship resilience, which is the key approach taken in this work.

The MassBrowser System: We have designed and implemented a new circumvention system, called MassBrowser, that aims at addressing the weaknesses of prior designs being based on the SoP principle. That is, MassBrowser aims at offering reliable blocking resistance while providing practical QoS and low operational costs. MassBrowser is a volunteer-run proxy-based system: it leverages normal Internet users with access to the free Internet, which we call *Buddies*, to proxy censored web traffic for censored users, i.e., *Clients.* The key to the resilience and QoS of any volunteer-based circumvention system like ours is to have a balanced ratio of proxying capacity to circumvention bandwidth demand. Towards this, we leverage the SoP principle to (1) optimize the proxying load on Buddies by using CacheBrowsing [29] and other *selective proxying* mechanisms introduced later, and, (2) encourage volunteer participation by giving Buddies full control and transparency over what they proxy. A central component of MassBrowser is a hard-to-block *Operator* service that oversees and enforces MassBrowser's key functionalities, particularly, by strategically matching Clients to Buddies based on the preferences of Buddies and the demands from Clients.

The following summarizes the intuitions behind MassBrowser's properties, which will be extensively discussed throughout the paper: **QoS and Cost:** MassBrowser combines several techniques including CacheBrowsing [29], selective proxying, and Domain Fronting [19] to optimize the QoS of circumvention connections while minimizing its operational costs. As shown in Section 8.2, we estimate the total cost of deploying MassBrowser to be no more than *$0.001 per active client per month.* **Blocking resistance:** MassBrowser's selective proxying not only optimizes QoS, but is also aimed at attracting a larger pool of Buddies by providing them full control and transparency over what they proxy (we support this claim by performing a user survey). Blocking MassBrowser's Buddies causes censors collateral damage as the Buddies are normal Internet users who frequently change network locations and connect from behind NAT. (i.e., to block a NATed Buddy, the censors will need to block the Buddy's subnet) To make an analogy, blocking MassBrowser Buddies is equivalent to blocking (the impractically expensive) domain fronted proxies. We also use state-of-the-art circumvention techniques to protect MassBrowser's Operator against blocking.

Deployment: MassBrowser is currently in the beta release mode, and we have implemented cross-platform end-user GUI software for Client and Buddy users with minimal technical background. We have been testing MassBrowser's performance for several months using volunteer clients from inside censored countries. Like any other volunteer-based system, MassBrowser will make a real-world impact only with wide adoption by volunteers who run MassBrowser Buddies. To come up with recommendations towards encourage wide volunteer participation, we perform *the first* user study on the willingness of Internet users in voluntarily helping circumvention technologies. The results of our user study suggest that *an encouragingly significant fraction of Internet users are willing to help censored users voluntarily—if they have full control over what is proxied through them.* We build MassBrowser's software as advised by the findings of our user survey, e.g., by allowing volunteers to whitelist the categories of websites they are willing to proxy, and the bandwidth they are willing to devote.

Privacy Guarantees: For normal clients, MassBrowser provides the same level of privacy as public VPNs/proxies and centralized systems like Lantern and Psiphon. Therefore, a Buddy can infer the Internet destinations of its clients, as well as their communication contents for non-HTTPS destinations (fortunately, major content providers such as news and social networking services offer HTTPS). On the other hand, a MassBrowser client can optionally compromise her QoS for stronger privacy properties. Specifically, our implementation of MassBrowser supports *connecting through Tor* for users who need anonymity in addition to blocking resistance (at the expense of a degraded QoS). This will tunnel a Client's Tor traffic through a Buddy who has opted to serve as a Tor bridge. Therefore, MassBrowser's Buddy software can be used as a pluggable transport [52] by Tor bridges. We evaluate MassBrowser's cost of operation when used as a Tor pluggable transport, showing that it is *drastically cheaper* than meek [44], while both offering similar blocking resistance properties (both meek and MassBrowser aim at increasing the censors' collateral damage by making use of shared IP addresses).

Summary of Contributions: In summary, we make the following main contributions:

1. We have designed a new circumvention system, MassBrowser, with the objective of addressing practical weaknesses of existing designs, particularly, blocking resistance, QoS, and operational costs.

2. We have performed the first user study on the willingness of normal Internet users in helping circumvention systems.

3. We have implemented and deployed MassBrowser as a fully operational system. We have used the findings of our user study to build a usable GUI software for clients and volunteers. Our software is hosted at `https://massbrowser.cs.umass.edu`.

2 Background on Circumvention Systems

Internet censorship is undoubtedly the biggest threat to the freedom of speech, ideas, and information across the globe [22]. To help censored users regain open access to the Internet, researchers and practitioners have designed and deployed an arsenal of tools known as *circumvention systems* [10, 14, 14, 15, 31, 32, 38, 41, 46, 50, 60, 61, 72, 74, 80]. Censorship authorities utilize their censorship technology to prevent the use of such censorship circumvention technologies by their citizens [35, 36, 57, 77], i.e., they block circumvention systems. In the following, we overview the major classes of circumvention systems and their weaknesses.

Proxy-based Systems The most common approach used by circumvention systems is to run network proxies outside the censorship region, and use them to relay the traffic of censored users to censored Internet destinations. Many in-the-wild circumvention systems such as Tor [16], Psiphon [53], Lantern [39], and VPN services [49, 51] deploy circumvention proxies in different ways to help censored users. Most circumvention systems [39, 49, 53, 62] use simple, single-hop proxies, while others [16, 52] use more complex models for proxy deployment. Tor, in particular, has implemented various *pluggable transports* [50, 52] to further hinder blocking by obfuscating the characteristics of Tor traffic.

Domain Fronting Domain fronting [19] is a blocking-resistant approach for setting up circumvention proxies. In this approach, the circumvention proxy is hosted on shared-IP infrastructures such as content delivery networks (CDNs), App Engines, and Cloud Computing services. Therefore the domain-fronted proxy will share its IP address with other, oblivious services making any censorship attempt susceptible to collateral damage. For instance, blocking a domain-fronted proxy hosted on a CDN requires the censors to block all the web content served by that CDN. CloudTransport [9] is an older variation of domain fronting, in which proxies are run over shared cloud storage services. Recently, several major content providers, including CloudFlare [12], Google [26], and Amazon [6], have started to disable or interfere with domain fronting, presumably in the fear of losing their market inside censored countries.

CacheBrowsing CacheBrowsing [29, 81] is a technique to fetch CDN-hosted censored content directly from CDN edge servers with no need to use circumvention proxies. To do so, various bootstrapping mechanisms are used to enable a censored client locate the CDN edge servers hosting her censored content of interest. CacheBrowsing is significantly cheaper [29, 45] than domain fronting since the CDN expenses are paid by the publishers of the censored content, not the circumvention operators. On the other hand, CacheBrowsing has a more limited scope as it can only be used to unblock certain censored content, i.e., those hosted on CDNs. In this paper, we leverage CacheBrowsing as a technique to optimize load on circumvention proxies, but not as a standalone circumvention system.

Protocol Tunneling Several circumvention proposals suggest to tunnel traffic through popular Internet services that are unlikely to be entirely blocked by the censors. For instance, FreeWave [32] tunnels circumvention traffic through VoIP services like Skype, and CovertCast [43] tunnels traffic through video streaming services. Alternatively, Rook [65] and Castle [28] tunnel traffic through gaming applications, and Sweet [34] tunnels through email communications. To block a tunneling circumvention system, the censors will need to block the oblivious service being used for tunnel, which has significant collateral damage to the censors [24]. On the downside, tunneling circumvention systems offer impractical QoS (e.g., high latencies and low bandwidth) due to the limitations imposed by their hosting services.

Decoy Routing Decoy routing aims at defeating IP address blocking by integrating circumvention software into the routing infrastructure [31, 37, 48, 80]. In decoy routing, censorship circumvention is implemented with help from a number of friendly Internet autonomous systems, called *decoy ASes*. Each decoy AS modifies some of its routers (e.g., its border routers) such that they deflect the Internet traffic of censored users to the blocked Internet destinations requested by the users. By design, decoy routing

Table 1: Weaknesses of major types of circumvention systems

Category	Easily blocked	Costly	Poor QoS	Deployability
Proxy-Based	●	◐	●	○
Domain Fronting	○	●	○	○
CacheBrowsing	○	○	●	○
Tunneling	○	◐	●	◐
Decoy Routing	○	◐	○	●

defeats IP address blocking, however, it is prone to particular routing-based blocking attacks known as RAD [33,47,55]. Requiring deployment by a number of in-the-wild ISPs is a major obstacle to the real-world deployment of decoy routing systems.

2.1 Weaknesses of Existing Systems

Here, we summarize the main weaknesses of existing circumvention systems, as summarized in Table 1:

1) Easy to block: Proxy-based circumvention systems, which encompass the majority of in-the-wild systems like Tor, Psiphon, and VPN services [49,51] can easily get blocked by the censors who enumerate their limited, small set of proxy IP addresses [57,59,75,77]. The censors can also use more advanced techniques like traffic analysis and active probing to block various kinds of circumvention systems [23,30,55,57,59,77].

2) Costly to operate: As introduced earlier, domain fronting aims at resisting IP address filtering by setting up proxies on shared-IP platforms such as CDNs, App Engines, and Cloud services. However, due to the prohibitively high costs of domain fronting [45], domain fronting is not used for circumvention proxying at scale, and recent proposals suggest to use domain fronting only for circumvention signaling, but not for proxying [56]. Several protocol tunneling systems [9] similarly need to some pay service providers for using their service, and decoy routing services require large investment in order to be deployed by Internet ISPs [47].

3) Poor QoS: Proxy-based circumvention systems like Tor suffer from low quality of services (e.g., high latencies) due to high congestion on the proxies. Various factors contribute to such congestion, most importantly the small number of proxies compared to clients, as well as the use of circumvention system by many clients for accessing bandwidth-extensive content such as copyright infringed multimedia content. Tunneling circumvention systems like FreeWave [32], Sweet [34], and CoverCast [43] offer low bandwidth and high latencies to the clients as they are constrained by the quality of service of their host services. CDNBrowsing systems [29,81] offer good latencies but can only be used to browse specific types of censored websites.

4) Hard to deploy: Some of the circumvention systems proposed in the literature are impractical to be used at large scale, despite offering reasonable blocking resistance and QoS. For instance, decoy routing systems [31,37,48,80] require wide adoption by Internet ISPs, and tunneling systems [32,34,43] can be trivially disabled by the third-party service providers they use for tunneling.

3 Sketch of our Approach

In this section, we present the key ideas behind the design of MassBrowser.

3.1 The Separation of Properties (SoP) Principle

We base the design of MassBrowser on the *separation of properties (SoP) principle* in order to overcome the shortcomings of existing circumvention solutions. The SoP principle states that *the key feature targeted by a circumvention system must be blocking resistance, and additional properties such as anonymity and browsing privacy should be provided as optional features to the users.* The SoP principle is based on the real-world observation [11,20,27,66,67] that the majority of censored users are solely interested in blocking resistance, e.g., to be able to access blocked news articles or to be able to communicate through blocked social networks; however, the *majority* of the censored users are not seeking properties like anonymity [79]. Our claim is supported by the vast popularity [66,67,78,79] of "public" VPNs, "public" HTTP proxies, and centralized circumvention systems like Lantern [39] and Psiphon [53] when compared to privacy-preserving solutions such as Tor (e.g., an estimated 31% of Chinese users have used VPNs in 2017 [79] compared to Tor's total 2 millions daily users).

Note that while systems like VPNs and HTTP proxies do not aim for anonymity/privacy, they *do not leverage* the SoP principle in optimizing censorship resilience. By contrast, we use SoP to optimize the blocking resistance performance of MassBrowser while offering practical QoS and cost of operation. Particularly, the SoP principle allows us to run single-proxy circumvention connections, which improves the QoS-cost tradeoff. Also, the principle allows us to restrict the use of our circumvention proxies to accessing censored content *only*. This not only reduces congestion on the proxies (therefore improving the QoS-cost tradeoff), but also increases the potential number of volunteer proxies by significantly reducing the legal consequences of running circumvention proxies, which has been a major issue for general purpose circumvention systems like Tor [7, 54].

For censored users who need additional properties like anonymity and browsing privacy, they can use MassBrowser to connect through (possibly censored) privacy-preserving tools like Tor. Particularly, our implementation of MassBrowser allows Tor clients to use MassBrowser Buddies as entry gateways to Tor, i.e., MassBrowser Buddies who opt in to proxy Tor traffic will serve as Tor bridges.

3.2 High-Level Design of MassBrowser

Figure 1 shows the high-level architecture of MassBrowser. MassBrowser is a volunteer-run proxy-based system: it leverages normal Internet users with access to the free Internet, which we call *Buddies*, to proxy censored web traffic for censored users, i.e., *Clients*. The key to the resilience and QoS of any volunteer-based circumvention system like ours is to have a balanced ratio of proxying capacity to circumvention bandwidth demand. Towards this, we leverage the SoP principle to (1) optimize the proxying load on Buddies by using several techniques including CacheBrowsing [29] and *selective proxying* (as will be introduced) and, (2) encourage volunteer participation by giving Buddies full control and transparency over what they proxy. A central component of MassBrowser is a hard-to-block *Operator* service that oversees and enforces MassBrowser's key functionalities, particularly, by strategically matching Clients to Buddies based on the preferences of Buddies and the demands from Clients.

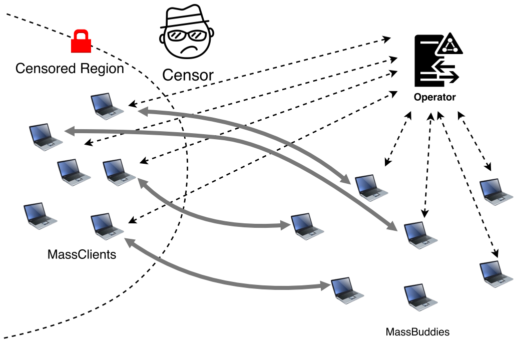


Figure 1: The main architecture of MassBrowser.

We will provide further details about these components and their interactions throughout the paper.

3.3 How MassBrowser Addresses Circumvention Issues

In the following, we summarize how MassBrowser aims at addressing the major circumvention issues discussed in Section 2.1. This will be further expanded later on.

1) Blocking resistance: As discussed earlier, proxy enumeration is the most common technique used by the censors to block proxy-based circumvention systems. Proxy enumeration is feasible in practice due to two reasons; first, the small number of proxy IP addresses used by typical circumvention systems enables the censors to enumerate all the IPs within a short interval [77]. Second, typical circumvention proxies use *dedicated* IP addresses that once identified can be blocked with no collateral damage. Domain fronting defeats IP blocking by using shared IP addresses, however is prohibitively expensive as a scalable solution. Our approach is to deploy a large number of volunteer proxies who (similar to domain fronting) share their (frequently changing) IPs with other Internet entities residing in the same NATed networks. Therefore, blocking MassBrowser's Buddies causes censors collateral damage as the Buddies are normal Internet users who frequently change network locations and connect from behind NAT. (i.e., to block a NATed Buddy, the censors will need to block the Buddy's subnet). Note that the key to MassBrowser's blocking resistance is recruiting a large number of volunteer proxies. We rely on the SoP principle to encourage volunteer proxying by (1) offering volunteers full control and transparency over what they proxy, and (2) optimizing proxying load on the Buddies.

2) Cost of operation: Similar to domain fronting [19] and CloudTransport [9], MassBrowser makes use of shared IP addresses to defeat IP enumeration. By contrast, MassBrowser is significantly cheaper to operate as the voluminous circumvention traffic is proxied through volunteer proxies. Also, while MassBrowser's Operator is implemented as a domain-fronted service to resist blocking, it only cost MassBrowser an estimated *$0.001 per active client per month* due to the small volume of its signaling traffic.

3) QoS: MassBrowser combines several complimentary techniques to offer a high QoS. First, it leverages CacheBrowsing [29] to minimize the traffic load on the volunteer proxies. Second, being based on the SoP principle, MassBrowser uses single-hop proxies for its connections (for the majority of the users who do not demand anonymity), and restricts the use of proxies to censored content. Third, MassBrowser aims at recruiting a large number of Buddies by providing them full control and transparency, which will impact the QoS of the connections.

4) Deployment feasibility: Unlike approaches like decoy routing systems [31, 37, 80] and tunneling systems [32, 34, 43], MassBrowser does not require cooperation/deployment from third-party Internet operators. Also, while MassBrowser's Operator is hosted as a domain-fronted service, it can be deployed using any low-bandwidth, high-latency covert communication mechanism [32,34] if domain fronting is widely disabled [6,26]. We have built user-friendly GUI software for both volunteers and clients to encourage wide scale adoption.

3.4 Comparison to Other Volunteer-based Systems

MassBrowser is not the first circumvention design to leverage volunteer proxies run by normal Internet users. In the following, we compare MassBrowser to such alternatives.

uProxy [62]: uProxy (currently, deprecated [62]) is another proposal to use volunteer Internet users as proxies for censored users. uProxy's original design [64] used the WebRTC protocol to connect a censored user to a volunteer proxy with an installed Chrome plugin. The uProxy project lately shifted towards using Shadowsocks [63] for connecting users to servers. uProxy did not use any central operator as in MassBrowser; instead, a uProxy censored user was supposed to know a friend outside the censorship region to act as her proxy. That is, uProxy would enable clients to set up "private" proxies, very much similar to private VPNs. We believe that this is not a scalable solution, as many censored users do not have close friends with access to the free Internet to help them.

FlashProxy [18, 21]: FlashProxy (currently, deprecated [21]) suggested to use volunteer websites to recruit ephemeral proxies. The volunteer website would load a particular JavaScript on each of its visitors, turning them into ephemeral proxies for censored clients. Even though a FlashProxy volunteer website would present a banner to its visitors informing them of the process, the visitors had no way to opt out except by refraining from visiting that website. We believe that high-visitor websites are unlikely to become volunteers as this may decrease their visitors. Additionally, the censors may retaliate by simply censoring (or even attacking) the volunteer websites.

Snowflake [56]: Snowflake is the successor of the FlashProxy project and uses some of the core communication protocols of uProxy [64], e.g., its WebRTC communication schemes. Similar to FlashProxy, Snowflake converts the visitors of some volunteer websites into circumvention proxies by loading a JavaScript. Therefore, we argue that a major challenge to Snowflake is adoption by volunteer websites: a volunteer website may get the target of censorship or cyberattacks by the censors, and therefore we do not expect adoption by major websites. Note that deployment by low-visitor websites does not help since the number of the proxies is proportional to the number of the visitors to the volunteer websites. Also, similar to Flashproxy, users in Snowflake have no way to opt out except by refraining from visiting the volunteer websites. By contrast, in MassBrowser we use Internet users to *knowingly* and voluntarily proxy traffic for censored users. Also, instead of using volunteer websites that turn their visitors into proxies, we use a hard-to-block central entity (the Operator) to strategically matchmake clients and volunteer proxies. MassBrowser implements various traffic optimization techniques and selective proxying to encourage volunteer proxying by respecting their preferences.

VPNGate [49]: VPNGate is a network of volunteers running VPN software open to the public. The VPNGate system maintains the list of all volunteer VPNs, and publishes the list on its webpage [70] for the interested clients. Unfortunately, VPNGate does not employ effective mechanisms to resist blocking, and therefore it is trivially blockable by the censors. The VPNGate website contains fake VPN IP addresses to prevent the censors from blacklisting the VPN IPs in bulk, however, the censors can easily identify and ignore such fake IPs by trying to connect to them through VPN protocols. In fact, the majority of VPNGate proxies appear to be currently blocked in China [68,69]. By contrast, in MassBrowser a blocking resistant Operator component establishes the connections between clients and proxies, preventing the censors from enumerating the proxies. Even if the censors enumerate MassBrowser's Buddy IPs, they can not block them without collateral damage as such IPs are NATed IPs with ephemeral port numbers, i.e., they change their port numbers for *every connection*. Additionally, MassBrowser deploys traffic obfuscation to defeat traffic analysis, while VPNGate's VPN traffic is trivially detectable at the network layer. As another distinction, MassBrowser employs various selective proxying techniques to optimize traffic load on volunteer proxies.

4 User Survey on Circumvention Participation

In addition to our MassBrowser, several recent proposals for circumvention work by using volunteer proxy operators [18, 49, 56]. While the key factor to the success of such systems is adoption by (a large number of) volunteers, there is no prior work evaluating the extent of support from volunteers. In this paper, we conduct *the first* user study on the willingness of uncensored Internet users in helping censored users through running circumvention software. We created an online survey questionnaire and distributed it among various groups of uncensored Internet users asking them about their interest and preferences in running circumvention software. In this section, we present the outcome of our survey.

Ethics. We received an IRB approval for our survey before distributing it among participants. The participants' data was collected and processed anonymously and voluntarily, and was stored on secure computer servers.

4.1 Survey Participants

We distributed our online survey among four groups of participants (a total of 300 participants):

1. `CS`: We advertised our survey among the members of a computer science department's social group. The participants are computer science students, faculty, and researchers working in various areas of computer science, all from the same US institution.
2. `OSN`: We distributed our survey on an online social networking platform. We expect the participants to be from diverse ethnic/technical backgrounds, though we did not request their fine-grained background information to keep the survey results anonymous.
3. `XCensored`: We distributed our survey among a group of people who are originally from a major

censoring country, but are currently living in the US The members of the group come from diverse educational backgrounds (mostly non-CS).

4. `MT`: We distributed our survey on Amazon Mechanical Turk (MTurk). The participants are all from US with diverse ethnic/technical backgrounds. Unlike previous groups who participated in the survey voluntarily, we paid our `MT` participants.

Table 4 (Appendix D) shows the demography of the participants for each of the surveys. Note that providing the demographic information was optional, therefore we did not receive the demographic information from all participants. We believe that our survey participants come from a diverse range of backgrounds to well represent broader Internet users.

4.2 Survey Format

The full survey questionnaire is included in Appendix C. We introduced the circumvention software as follows: "Suppose that there is a software called Helper that when you install on your laptop/desktop computer, it will assist the Internet users in censored countries to get around censorship". We also told the participants "Assume that someone you trust guarantees that the Helper software will not make any harm to your computer or your network. Also, the use of Helper is transparent to you and does not interfere with your work". Then, we asked participants about their willingness in running the "Helper" circumvention software and their preferences, as discussed in the following.

4.3 Survey Results

Willingness to Install and Run the Software. We first asked each participant if they were willing to install and run the Helper software—for free. We told them that "running Helper does not cost you anything, but also does not earn you money." We also mentioned "you can completely control the use of Helper (as will be asked in the follow up questions)". If a participant expressed her unwillingness, we asked her if she would participate if she got paid.

Figure 2 shows the aggregation of the responses. As we can see, *a significant fraction of surveyed participants (51%) expressed their willingness to run the circumvention software completely voluntarily (i.e., for free), if they trust the software to be harmless.* This is an encouraging finding for volunteer-based circumvention systems: with adequate security and safety protections, one can expect to recruit a decent number of volunteer circumvention helpers. Also, about 36% of people who were unwilling to participate for free (≈ 17% of all participants) expressed their willingness to participate if they got paid.

Comparing the responses from different groups of participants we see varying participation interests. Particularly, the `MT` participants pose to be less interested in helping a circumvention system for free; we believe the `MT` results present a pessimistic estimation of the general population of users, since Mechanical Turk users have stronger financial motives compared to the general public (i.e., they take MT jobs to get paid instead of voluntarily). Nonetheless, even our `MT` results are encouraging enough, i.e., above 50% free participation. Also, we find that the `XCensored` participants are less willing to participate than the other two groups. We speculate this to be due to their family/business bonds with the censoring countries. Finally, we see the most willingness from participants with more familiarity with technology (i.e., the `CS` and `OSN` groups).

Content Type Preferences. We asked the willing participants about the type of content they are willing to proxy for censored users. We warned them that "some censored users (whom you don't know) will use your computer to connect to censored Internet websites. So your Internet provider may assume that you are browsing those websites yourself. What kind of websites do you feel comfortable (and allow) to be proxied through your computer by censored users."

Figure 3 shows the results for various categories of censored content. We observe that only a small fraction of participants (about 24%) are willing to relay *any* type of traffic, and only about 18% of the participants are willing with proxying *any legal websites*, however, the rest of the participants (58%) prefer to specify the type of traffic they are willing to proxy.

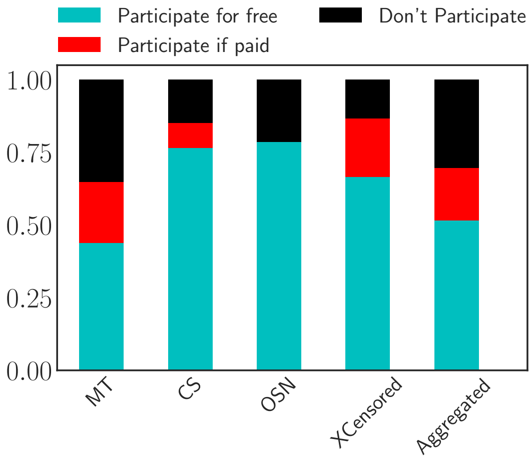


Figure 2: Survey participants' willingness to install and run a software that helps censored users.

For such participants, we see that Video Streaming websites are the least accepted while News and Scientific websites are the most accepted categories.

Censored Country Preferences. We asked the willing participants if they preferred to help censored users from any specific countries. As shown in Figure 4, the majority of the willing participants ($\approx 71\%$) had no particular preference on the ethnicity of the censored users they were helping.

Bandwidth Devotion Preferences. We asked the willing participants how much of their "unused bandwidth" they are willing to allocate to the circumvention software. As shown in Figure 5, the majority of the participants are willing to devote substantial fractions of their "unused" bandwidth for circumvention. We particularly see that around 50% of participants are willing to donate more than half of their unused bandwidth for circumvention.

4.4 Incorporating the Results Into MassBrowser

We use the findings of our survey to guide the design of our volunteer-based circumvention system, MassBrowser.

- We observe that a significant fraction of our survey participants are willing to install and run a circumvention software to help censored users (given guarantees on their safety and security). We find this an encouraging finding for emerging volunteer-based circumvention systems like MassBrowser. The survey also shows that most of the willing participants are willing to help for free, therefore we do not see an immediate need for incentivizing mechanisms for MassBrowser.
- Our participants were told that someone they trust guarantees their security and safety. Towards this, we have released MassBrowser's code as open source software, and we are undergoing third-party *code review* by a reputable organization.
- We observe that the willing volunteers have various reservations about how they are willing to proxy circumvention traffic, particularly on the type and volume of the content they proxy. Therefore, we deploy MassBrowser in a way to enable its volunteers adjust how the software runs on their computers. We believe that the sparsity of volunteers in popular systems like Tor is due to the lack of such controls and guarantees, especially given recent incidents for Tor exit operators [7, 54].
- Our survey shows a greater interest from technology-aware participants in voluntarily helping circumvention software. Advised by this, we have reached out to technology-aware Internet users to help MassBrowser during its bootstrapping phase.
- To enable participation from a wide spectrum of volunteers, we build a simple, user-friendly GUI interface for volunteer proxy operators.

5 MassBrowser's Threat Model

We assume that MassBrowser Clients are located inside censoring regions, and Buddies are users residing in non-censoring regions with open Internet access. The censorship authorities monitor the Internet

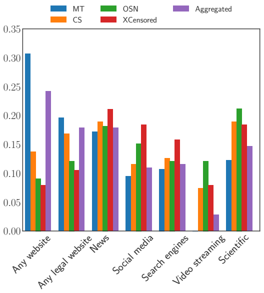


Figure 3: Participants' willingness to proxy different types of content.

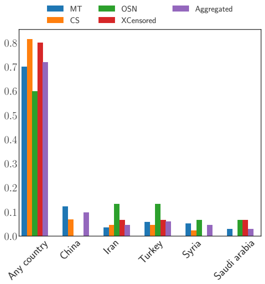


Figure 4: Participants' willingness to help censored users from different countries.

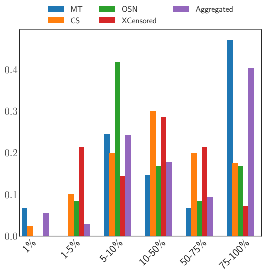


Figure 5: The fraction of unused bandwidth that participants are willing to devote to a circumvention software

communications of the censored Clients, and are able to block or interfere with any connection from such Clients to Internet destinations. Censors are also able to act as Clients or Buddies in order to gain information about the system and to disrupt the system to the best of their ability. However, we assume that censors are not capable of tampering with users' devices (e.g., installing monitoring softwares on their devices), as this will disable any privacy-enhancing tool.

We assume the players in our system to be *rational*. A rational censor tries to minimize the costs and collateral damages incurred by its actions, such as interfering with benign, popular Internet services. Buddies are rational in that they are willing to help censored users as long as this does not pose any risks to themselves. For example, a Buddy will not let Clients use her device to deploy network attacks (e.g., port scan, sending spam email) or to access controversial destinations that will get the Buddy into trouble.

We also assume that the *censors do not penalize normal users for the sole act of using a circumvention software*, unless the websites accessed are directly related to major criminal offenses. Although using circumvention tools is considered illegal in many censoring countries, penalizing Internet users solely for using a circumvention software is very rare in most countries [11] (instead, the censors have penalized people who *operated* a circumvention software for others [1, 3]). For example, as of 2017, Facebook has over 17 million users from Iran accounting for over 20% of the population [8], despite it having been blocked for more than 8 years. The fact that users are willing to provide public information on a blocked website confirms the negligible risk of using circumvention software in such countries. Our threat model assumes that the censored clients are aware of, and accept the (negligible) risks of using a circumvention software, and therefore users who need more protection like journalist should use MassBrowser in combination with an anonymity system like Tor.

6 MassBrowser's Design Decisions

In this section, we detail our design decisions aimed at addressing each of the issues discussed in Section 2.1.

6.1 Blocking Resistance

We use the following core techniques to provide a high blocking resistance in MassBrowser.

Use of shared, dynamic proxy IPs to resist IP enumeration As MassBrowser proxies are run by normal Internet users, blocking them is costly and prone to collateral damage. First, a typical Buddy volunteer will most likely have a NAT IP address, therefore sharing a public IP address with other users/services in the same network. For instance, a Buddy connecting from a coffee shop will share a public IP with other users in the area (we will describe how MassBrowser enables connections despite NAT). Additionally, a typical Buddy will frequently change IP addresses, e.g., by moving across networks, amplifying the collateral damage. Second, by employing various social engineering techniques, described later, we hope that MassBrowser will attract a large number of volunteer proxies making IP enumeration unreliable and costly (in addition to its high collateral damage).

Traffic Obfuscation and Encryption All MassBrowser communications are encrypted to prevent deep-packet inspection. Additionally, MassBrowser deploys traffic obfuscation mechanisms to remove protocol fingerprints and prevent censors from detecting MassBrowser traffic based on traffic characteristics like packet timings and sizes.

Domain Fronting the Operator MassBrowser's Operator runs as a domain fronted service [19]. As discussed earlier, a domain fronted service runs behind a network infrastructure with shared IPs (e.g., CDNs), therefore blocking it will cause significant collateral damage to the censors. Although domain fronting is a relatively expensive technique, the costs of domain fronting MassBrowser's Operator is very low due to the small volume of the control traffic generated by the Operator, as shown in Section 8.2. Note that while MassBrowser's Operator is hosted as a domain-fronted service, it can be deployed using any low-bandwidth, high-latency covert communication mechanism [32, 34] if domain fronting is widely disabled in the wild [6, 26].

Other Privacy Properties: Being based on the SoP principle, MassBrowser is optimized around blocking resistance, not anonymity or browsing privacy. In Section 9 we will thoroughly discuss MassBrowser's privacy guarantees against various adversaries.

6.2 Optimizing Cost and QoS

As discussed earlier in Section 3, blocking resistant circumvention systems suffer from either low QoS or high cost of operation (or both). We argue that the main reason for the poor QoS/high cost of existing circumvention systems is the extreme disproportion between available proxying throughput and the bandwidth demand from censored clients. We therefore take the following two complimentary approaches to alleviate such disproportion.

Optimizing load on proxies through selective proxying: We use the following techniques to minimize the traffic load on MassBrowser proxies.

a) Whitelisting censored content only: Existing circumvention tools like Tor and VPNs tunnel *all* network traffic of a censored client through circumvention proxy, including censored and non-censored content. This is done in Tor to provide anonymity on all connections, but even non-anonymous tools like VPNs, Lantern, and Psiphon tunnel all traffic through circumvention proxies for the ease of operation. We believe that this is one of the key reasons constituting to high bandwidth pressure on in-the-wild circumvention proxies (causing their low QoS). We evaluated the list of top bandwidth-consuming domains provided to us by a major non-anonymous circumvention tool[1] for the day of Feb 21, 2008. Our evaluation finds that 48% of the proxied traffic belongs to websites that are *not* censored in Iran (total proxied traffic is 3.56 TB).

Tunneling non-censored content through a circumvention system not only puts additional burden on the proxies, it also lowers the quality of service for most of the non-censored websites, e.g., a Chinese user will have to access a (non-censored) China-based website through a US-based proxy, therefore increasing the latency. Basing our design on the SoP principle, we restrict the use of MassBrowser Buddies to censored-content only. Therefore, our Client software only proxies censored content through Buddies and retrieves non-censored content directly with no proxy, and the Buddies deploy whitelists to proxy only censored content.

b) CacheBrowsing: MassBrowser uses a recent circumvention technique called CacheBrowsing [29, 81] to further minimize the load on the proxies. As introduced in Section 3, in CacheBrowsing a client directly fetches a censored object hosted on CDN from the hosting CDN's edge servers, without using proxies. However, a limitation of CacheBrowsing is that it can only retrieve censored content hosted on a CDN and

[1] We do not disclose their identity per their request.

accessible through HTTPS, therefore it can not be used as a standalone circumvention system. We integrate CacheBrowsing into MassBrowser's client software. That is, a MassBrowser client will fetch the CDN-hosted censored content directly from CDNs, and only use MassBrowser Buddies for the censored content not hosted on a CDN. Based on our analysis, this saves 41% of bandwidth on the Buddies for Alexa top 1000 websites.

c) Strategic proxy assignment to prevent DoS by Sybils: In MassBrowser, clients discover Buddies and connect to them with help from the Operator. MassBrowser's Operator uses a proxy assignment mechanism, described later, to prevent the censors from learning a large fraction of Buddies. Note that, even if the censors can enumerate all Buddy IPs, they can *not* block the discovered (NATed) Buddies due to the collateral damage, but they can possibly try to consume their circumvention throughput.

Incentivizing volunteers to recruit more proxies: The QoS of a proxy-based circumvention system critically depends on the number of its proxies. We use the following approaches to increase the number of volunteer proxies. We envision a large fraction of MassBrowser Buddies to be from typical Internet users with little technical background. We therefore design a GUI-based client software for Buddies to offer a user-friendly experience, transparency, and full, fine-grained control over what they proxy. Our Buddy GUI offers the following features.

a) Imperceptible operation: Our Buddy GUI runs imperceptibly and does not interfere with the volunteer's normal activities. The volunteer user will only need to perform a one-time installation and setup of the relay software, and may then let it operate until she needs to adjust her preferences.

b) Transparency on usage: Our Buddy GUI offers the volunteer with information on how the proxy is being used.

c) Enable relays to limit proxied bandwidth: The Buddy software enables a volunteer Buddy operator to specify how much bandwidth she is willing to donate to MassBrowser. Even a small donated bandwidth can help MassBrowser clients due to the bandwidth minimization mechanisms discussed above.

d) Enable relays to whitelist destinations: Our MassBrowser relay software enables a volunteer to proxy traffic only to Internet destinations she is comfortable with. A major set-back for volunteers is the potential legal consequences of relaying traffic to controversial destinations (such as those experienced by Tor exit relay operators [7, 54]). In MassBrowser, relays whitelist the categories of destinations they are willing to proxy traffic to, e.g., a relay can decide to relay traffic only to news websites or scientific websites.

e) Optional financial incentives Future versions of MassBrowser may incorporate financial incentives for volunteers, either as the form of a service like Bitcoin mining by clients, or monetary compensation. We leave the investigation of incorporating such economic incentives with MassBrowser to future work.

6.3 Deployment

Unlike some of the previous circumvention systems like decoy routing [31, 37, 80] and tunneling systems [28,32,34,65], MassBrowser's operation does not rely on any third-party operators like autonomous systems and services providers. MassBrowser has recently been released in beta version to limited number of users. We have built user-friendly GUI software for both Clients and Buddies for the major operating systems.

7 MassBrowser's Implementation Details

In Section 3, we introduced the high-level design of MassBrowser, and in Section 6 we discussed MassBrowser's design decisions. In this section, we will present more details on MassBrowser's implementation. Due to space constraints, more specific details about our code is presented in Appendix B.

7.1 Connecting Users Behind NAT

As MassBrowser Clients and Buddies are regular Internet users, most of them will likely be connecting to the Internet using NATed IP addresses. Therefore, an important challenge to MassBrowser's operation is enabling communication between NATed Clients and Buddies, i.e., MassBrowser needs to deploy *NAT traversal* techniques [42,76]. Typical NAT traversal techniques, however, may not be applicable for all transport protocols depending on the type of a peer's NAT, i.e., depending on how the underlying NAT maps local IPs to public IPs. Matthews et al. [42] perform a thorough analysis of different NAT deployments in the Internet and how NAT traversal techniques may apply to them. We categorize MassBrowser peers (i.e., Clients and Buddies) into three categories based on the type of their NATs.

TCP Reachable These are the peers with whom it is possible to initiate a TCP connection, either directly or via some existing NAT traversal technique.

UDP Reachable For such peers, we are not able to initiate TCP connections, but are still able to send UDP packets to them via some NAT traversal technique. These peers reside behind *Restricted NATs* as defined by Wing et al. [76].

Unreachable Such peers are located behind NATs that prevent the use of *any* NAT traversal technique. Wing et al. [76] classify these NATs as *Symmetric NATs*.

MassBrowser's Operator serves as a STUN server to discover the NAT type of each peer. The Operator then uses the discovered NAT type of the peers to match Clients and Buddies, and to decide which party should initiate the connection, as shown in Table 2. For any pair of a Client and a Buddy, they can communicate if at least one of them is reachable from behind NAT. As can be seen, when both of the peers are reachable, the Client initiates the connection. When both peers are UDP reachable, MassBrowser's software tunnels a TCP connection through an established UDP tunnel. If none of the peers are reachable, a MassBrowser connection can not be established between these peers, and therefore the Operator will *not* map an unreachable Client to an unreachable Buddy.

Note that MassBrowser's Operator *does not* deploy a TRUN server; a TURN server will need to proxy the connections between (unreachable) Clients and Buddies, which is significantly expensive and bandwidth-extensive for a free circumvention system like ours.

Table 2: Connection initiation for a matched pair of Clients and Buddies. If both the Client and Buddy are unreachable, the Operator will not match them together.

		Buddy		
		TCP-Reach	UDP-Reach	Unreach
Client	TCP-Reach	Client	Buddy	Buddy
	UDP-Reach	Client	Client	Buddy
	Unreach	Client	Client	✗

Additionally, a circumvention TURN server can easily get blocked by the censors unless it is deployed as a (prohibitively expensive) domain fronted service.

7.2 Assigning Buddies to Clients by the Operator

The Operator is in charge of coordinating Client and Buddy communications and providing Clients with online Buddies to use as relays. The Operator assigns Buddies to Clients with the following considerations.

Buddy destination whitelists Buddies can whitelist destinations they are willing to proxy traffic to based on their content types. The Operator actively maintains Buddy whitelist preferences. When a Client queries the Operator for new Buddies, the Operator will respond with Buddies that allow the intended destinations in their whitelists.

Buddy loads The MassBrowser system is a heterogeneous network composed of machines with varying processing powers and network bandwidths. The Operator approximates a Buddy's available throughput based on the bandwidth limit set by the Buddy owner, the number of active Clients assigned to that Buddy, Buddy's reliability over time. This is used to balance the load on Buddies when assigning Buddies to new Clients.

Parties' NAT types The Operator also considers the NAT types of the peers in matching Clients and Buddies, as described above.

Sybil attack protection As discussed in Section 9, a censor can *not* block the Buddies that she obtains from the Operator, nor can she identify their clients (since Buddy IPs are NATed). However, a resourceful censor may overload the identified Buddies in order to consume their circumvention capacity (i.e., DoS the Buddies). Note that this will be a costly DoS attack due to the symmetry between the load on the attacker and the target. Nonetheless, our Operator can deploy standard Sybil protection mechanisms against such an expensive DoS attack. We have particularly borrowed and implemented the Sybil protection mechanism used by rBridge [73]. This mechanism uses a reputation system for clients in order to provide them with new proxy information. Similar to rBridge, we have deployed an invitation-based mechanism for accepting new clients.

7.3 Client-side Optimization of Proxy Loads

As discussed in Section 6.2, MassBrowser deploys *selective proxying* to optimize the load on the Buddies. By contrast to in-the-wild circumvention systems that *naively* proxy everything through the circumvention proxies, MassBrowser inspects every network request individually and decides how to best handle that individual request. Figure 6 shows how MassBrowser Client implements such selective proxying. MassBrowser relays a request through Buddies only if it identifies the requested content to be censored for the Client. Furthermore, if the requested resource is CacheBrowsable [29], MassBrowser will fetch the content (either fully or partially [81]) directly from CDNs imposing no load on the Buddies, and will only proxy the non-CacheBrowsable components of the connection.

To perform such per-request targeted proxying, the MassBrowser Client must have a means of identifying censored and CacheBrowsable URLs. For this purpose, the Operator actively maintains a database of MassBrowser-supported websites along with detailed information about the different resources and components of each website. The MassBrowser Client software keeps a regularly-synced local version of this database.

Note that the Operator itself is deployed as a domain fronted service. All the communications between the Client and the Operator, such as those required for updating the local database, requesting Buddies, and NAT traversal, will be domain fronted and thus unblockable by the censors. On the other hand, the connections between Buddies and the Operator does not need to be domain fronted.

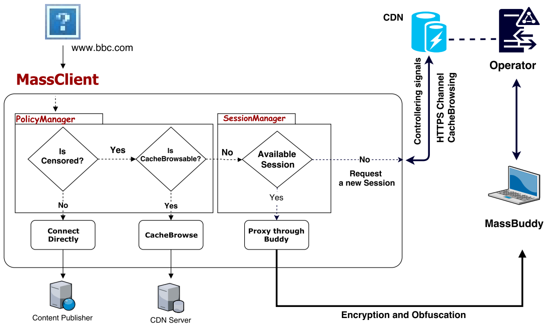


Figure 6: Optimizing proxying load by a MassBrowser Client through selective proxying

7.4 Content Whitelisting

As described above, a key load optimization mechanism employed by MassBrowser is to only proxy censored content through Buddies. Also, different Buddies have personalized preferences for the categories of traffic they are willing to proxy to. To be able to implement such functionalities, MassBrowser deploys content whitelisting to enable the Clients identify the content objects that should be proxied, and to enable the Buddies to enforce their restrictions.

In order to enforce such whitelisting policies, the Client's web browser delegates DNS resolution to MassBrowser's Client software (this requires disabling the browser's DNS caching); therefore, proxy destinations must be hostnames, not IP addresses. This enables distinguishing requests to different hosts that resolve to same (shared) IP addresses. For every web request from the client, the Client software looks up the requested destination hostname in its database of supported websites, identifies which website the hostname belongs to, identifies whether the website is censored, and determines the content types associated with that website. If the request is censored and the Client already has an open session with a Buddy that supports the required content's type, it will use the existing connection to proxy that request.

Otherwise, the Client will be assigned a Buddy who has whitelisted the category of the requested content.

On the other hand, each Buddy is in charge of performing DNS resolution for Client requests; therefore each Buddy is able to ensure that the Client is not violating the Buddy's destination restrictions.

To whitelist content for applications other than web browsers, the whitelisting mechanisms are tailored for such specific applications. In particular, Buddies that accept Tor traffic do so by whitelisting the IP addresses of all Tor public relays

7.5 Encryption and Traffic Obfuscation

In MassBrowser all of the communication between Clients and Buddies are encrypted in order to resist DPI attacks deployed by the censors. A matched pair of Client-Buddy encrypt their messages using a symmetric cipher with a shared secret key that they share through the Operator. Our implementation currently uses AES 256 for Client-to-Buddy encryption.

We also implement traffic obfuscation to protect MassBrowser's traffic against traffic analysis attacks [23,30,71]. Particularly, we have built a custom implementation of the *obfsproxy* [50] Tor pluggable transport tailored to work with our MassBrowser implementation. The obfuscation algorithm removes identifiable traffic patterns, making the Client-Buddy protocol look like benign peer-to-peer traffic, e.g., p2p gaming or file sharing traffic.

7.6 Communication Sessions in MassBrowser

We define a MassBrowser *session* to be a connection between a Client and a Buddy. Upon receiving a request from the browser, the Client checks whether the request can be handled with any of the currently active sessions the Client has, i.e., whether any of the connected Buddies will accept the request in their whitelisted categories. If no such session is found, the Client will need to ask the Operator to assign it a new session with a suitable Buddy that will accept the request.

The Operator will select a Buddy to assign to the Client and will notify both parties to establish a new session. Each session has the following attributes:

1. *Allowed content types:* This is the list of content types that the Client is allowed to obtain through this session.
2. *Shared Keys and Cipher Suite:* All communications between the Client and Buddy are encrypted with a shared key and cipher suite shared through the Operator.
3. *Obfuscation method:* In order to prevent fingerprinting attacks on the Client-Buddy communication protocol, the Operator may instruct the users to use one of the available obfuscation algorithms if the censoring region is known to deploy DPI attacks.
4. *Connection initiator:* Based on NAT type of the peers, the Operator will instruct one of the users to initiate the connection with the other using an appropriate NAT traversal technique, as described earlier.
5. *Expiration time:* Each session is only valid within a defined time period. The Client will have to ask to renew the session if he wishes to continue using it beyond the expiration time. This is to perform load balancing on Buddies over time.

The Operator will send the details of each new session to the corresponding Client and Buddy. The party who has been selected as the connection initiator will then attempt to establish a connection with the other party. The receiving party will keep the session in a list of pending sessions until either the connection is established or the session expires. Each session can only be used once, and both parties will notify the Operator once the session connection has been established. Figure 7 shows the messages involved in establishing a session, and how traffic is relayed between Clients and Buddies.

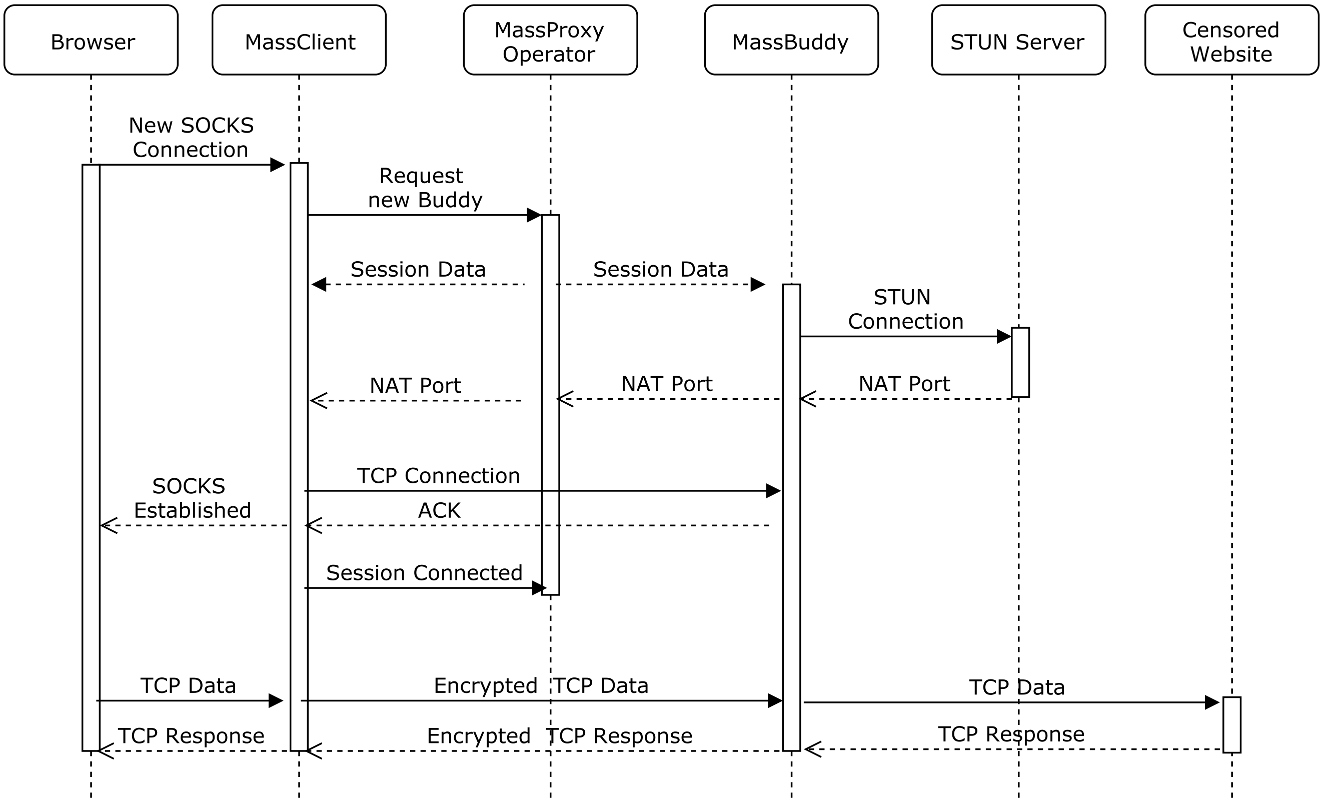


Figure 7: A MassBrowser communication session between a Client and a Buddy. In the shown example, the Client is selected to initiate the connection by the Operator.

8 Performance Evaluation

8.1 Buddy Bandwidth Contribution

Our analysis of the top 1000 Alexa website homepages [5] finds the average size of each webpage to be 2.4 MB. We found 41% of the generated traffic by these pages to be CacheBrowsable, which is very promising to us for load optimization (note that most of the CacheBrowsable webpages are partially CacheBrowsable [81], therefore MassBrowser needs to proxy only the non- CacheBrowsable components). Therefore, in order to load a typical page through MassBrowser the client will only need to proxy an estimated ≈ 1.4 MB through the Buddies. The Akamai State of the Internet Connectivity Report [4] estimates the Internet bandwidth of an average user living in the United States in 2017 to be 18.7 Mbps. Assuming volunteers will provide MassBrowser with 25% of their unused bandwidth (which is very likely based on our user survey in Section 4), an average Buddy in the United States will contribute 4.7 Mbps when not using the internet, which translates into a page load every 2.5 seconds. Also, recall that in MassBrowser, the bandwidth of Buddies is solely used for loading censored content.

8.2 Costs of Operation

Ensuring low operational cost is one of the primary design goals of MassBrowser. The (bulky) circumvention traffic of MassBrowser clients is handled by volunteer Buddies. Therefore, the only operational cost of MassBrowser is imposed by running the Operator. Recall that the Operator is deployed as a domain fronted service, i.e., hosted on a CDN, in order to allow unblockable access to the censored users. In this section, we show that while domain fronting is known to be prohibitively expensive for proxying [45], it imposes little costs on MassBrowser as it is only used for its control traffic.

There are three factors that contribute to the Operator's operational costs:

1) Number of Client-Requested Sessions Per Day: Each session established between a Client and a Buddy is capable of serving any volume of traffic to different destinations as long as they satisfy the content type restrictions imposed by the Buddy. Therefore, it is unlikely that a Client will require more than a few active sessions at any given time. Our evaluation of a typical Client shows that 20 sessions per day is sufficient for typical web browsing.

2) Size of Session Objects: Upon creation of a new session between a Client and a Buddy, the Operator will need to exchange some protocol messages to the two parties. The exchanged information is composed of a 500 byte fixed-size segment containing details about the IP addresses, ports, NAT types, connection initiator, secret key, and the session expiration date, along with a variable-size segment listing the content types that will be accepted on the session (each content type takes 12 bytes). Therefore, the overall traffic load on Operator for each session is $\approx$ 1000 bytes.

3) Size of the Webpage Database: The Operator maintains a database containing information on how to browse different censored websites supported by the Buddies. While the number of such unique domains for every website could be high, the database stores the domains in regex format, combining groups of similar domains with identical censorship information into single entries. The majority of the websites have at most 50 entries in Operator's database; given that each entry is around 1KB, each website will use at most 50KB in the database.

Based on these factors, we estimate Operator's operational costs, which is hosted over the Amazon AWS.

Cost of Running the Operator Servers: We estimated every user to request 20 sessions per day. For 10,000 users this requires 200,000 requests which would amount to an average of 2 requests per second. An AWS EC2 *t2.micro* instance, costing at about $0.015 an hour, will be sufficient for handling this load of requests generated by 10,000 users. *The monthly cost will amount to $0.0011 per user.*

Cost of Deploying on CDNs: We have hosted the Operator on the Amazon Cloudfront CDN. Amazon Cloudfront charges based on the volume of traffic, and the locations of the CDN edge servers used. Note that Operator's communications with Clients are *not* latency sensitive; therefore, it suffices for the Operator to use a cheap CDN service (we use a service with $0.01 per GB). As estimated above, each user will request 600 sessions per month, for which the Operator will need to send 600 KB of control data to the Clients; this costs *$0.00006 per user each month*. The user will also need to synchronize her local database with Operator, resulting in a one-time 50 KB data transfer for each supported website, which costs $0.0000005 per user for every website.

Comparing costs with meek: Meek [44] is a Tor pluggable transport that relays Tor traffic through domain fronted proxies to evade censorship. In order to operate, meek must proxy all of the users' traffic through CDN servers. As a result, unlike MassBrowser the costs of operating meek is proportional to the client's bandwidth usage. As we saw in the previous analysis, we estimate the cost for a MassBrowser user with 600 sessions per month to be $0.00006 each month using Amazon Cloudfront CDN, *regardless of the types of the websites browsed* (e.g., video streaming, news, etc.). If we assume each session to be just for one website load and each website to have an average of 2.4 MB (as we measured), then the same client using meek over Amazon Cloudfront CDN will cost $600 * 0.0024 * 0.01 = \$0.014$, which is over *200 times* the cost of the user on MassBrowser. Note that in real life each session will be used to browse multiple websites and may require higher traffic (e.g., for video streaming), therefore, the cost gap will be even greater in favor of MassBrowser.

Table 3: Average page load latencies for different website over Tor, MassBrowser as a Tor bridge, and MassBrowser alone.

Website	Tor (s)	MassBrowser + Tor (s)	MassBrowser (s)
Google.com	19.6	20.3	2.6
Youtube.com	27.3	25.6	6.3
Facebook.com	27.4	30.4	6.6
Baidu.com	7.5	10.1	1.7
Wikipedia.com	29.5	22.3	1.1

8.3 MassBrowser as a Tor Transport

As mentioned before, MassBrowser can be used as a Tor pluggable transport, i.e., a Client who needs anonymity can connect to a Buddy who whitelists Tor traffic. We measured the time to load the top 100 Alexa websites with Tor, using MassBrowser as a bridge for Tor, and using MassBrowser without Tor. We browsed each website 50 times over each setting and computed the average time to load the websites. Table 3 presents the load times for different websites. On average *loading each website on Tor takes more than 16 seconds longer than using MassBrowser*. Using MassBrowser as a Tor bridge does not significantly change the load times compared to using Tor with no pluggable transport; therefore, *MassBrowser's added latency on Tor is negligible*, making MassBrowser a suitable plug for Tor bridges.

9 Discussion of Privacy Guarantees

In this section, we discuss the privacy guarantees of MassBrowser's components. Please refer to Appendix A for potential questions not discussed here.

9.1 Client Privacy

A. Privacy against Buddies A MassBrowser Buddy imposes the same privacy threats to its Clients as a network observer, e.g., an ISP, on regular Internet users.

Anonymity against Buddies: As discussed earlier, providing client anonymity is not a design goal for MassBrowser based on the SoP principle. Therefore, a Buddy can learn the destinations being accessed by her connected Clients —this is similar to how a typical network observer (like an ISP or a transit AS) can learn browsing patterns of typical Internet users. Note that, like a normal Internet user, a MassBrowser client needing anonymity can use an anonymity system like Tor—through MassBrowser —(i.e., by connecting to Buddies that support Tor).

Confidentiality from Buddies: A Buddy will not be able to see its Clients' communication content for HTTPS destinations, which includes the majority of services hosting sensitive user data like social networking websites and search engines. A Buddy, however, will be able to see a Client's communication content to an HTTP destination, similar to how an ISP observes the HTTP traffic of its users. A MassBrowser Client can opt to use MassBrowser for HTTPS websites.

Surveillance by censor-run Buddies: A powerful organization that runs numerous Buddies for user surveillance is not different than a nation state or ISP wiretapping through Internet routers. Real-world observations over the years have shown that censoring governments tend to not penalize their users for the sole act of circumventing censorship. The risk is much less for MassBrowser Clients as, by design, MassBrowser Buddies do not allow connection to controversial websites with potential legal consequences (for such websites, the clients will need to use Tor through MassBrowser).

Identification by censors who know Buddies: The Buddies obtained by a censoring client from Operator can not be used to learn any information about the Clients who use these Buddies. This is because different Clients connecting to the same Buddy will make connections through different IP address and port combinations due to NAT.

B. Privacy against Operator Unlike traditional circumvention tools like Psiphon, Anonymizer, and Lantern, in MassBrowser the Operator of the circumvention system is separate from the proxying parties. Therefore, the Operator is not able to observe Client traffic. The Operator can only learn the categories of content a Client is willing to access.

9.2 Buddy Privacy

Privacy against Clients A Client using a Buddy will only learn the (ephemeral) *NATed* IP address of that Buddy, but no other information. As Client-Buddy assignments are performed by the Operator, a Client can not choose the Buddy to connect to.

Privacy against Operator The Operator will have access to a Buddy's preferences such as her whitelisted content types and specified bandwidth limits. A Buddy's IP address will also be exposed to the Operator, however similar to the Clients, this is the *NAT* IP address of the Buddy, which is also visible to any other web service the Buddy connects to on the Internet.

Acknowledgements

The project has been supported by the NSF CAREER grant CNS-1553301.

References

[1] Behind China's VPN Crackdown, A 'Game Of Cat And Mouse' Continues. https://www.npr.org/sections/alltechconsidered/2017/08/04/541554438.

[2] Django Web Framework. https://www.djangoproject.com/.

[3] Iranian regime arrests VPN software seller. https://www.ncr-iran.org/en/news/human-rights/13125-iranian-regime-arrests-vpn-software-seller.

[4] Akamai. State of the Internet Connectivity Report, Q1 2017. https://www.akamai.com/us/en/about/our-thinking/state-of-the-internet-report/global-state-of-the-internet-connectivity-reports.jsp.

[5] Alexa. Top Websites. https://www.alexa.com/topsites.

[6] Amazon threatens to suspend Signal's AWS account over censorship circumvention. https://signal.org/blog/looking-back-on-the-front/, May 2018.

[7] Dark net raids were overblown by police. http://www.bbc.com/news/technology-29987379.

[8] Azali, M. Infographic: Facebook Usage Statistics in Iran. http://techrasa.com/2017/08/16/infographic-facebook-usage-statistics-iran/.

[9] Brubaker, C., Houmansadr, A., and Shmatikov, V. CloudTransport: Using Cloud Storage for Censorship-Resistant Networking. In *PETS* (2014).

[10] Burnett, S., Feamster, N., and Vempala, S. Chipping Away at Censorship Firewalls with User-Generated Content. In *USENIX Security* (2010).

[11] Penalties For Using VPN In Various Countries. https://www.vpnunlimitedapp.com/blog/penalties-for-using-vpn/.

[12] Another Strike Against Domain Fronting. https://wills.co.tt/1746/another-strike-against-domain-fronting, Feb. 2017.

[13] Amazon Cloudfront CDN. https://aws.amazon.com/cloudfront.

[14] Defeat Internet Censorship: Overview of Advanced Technologies and Products. http://www.internetfreedom.org/archive/Defeat_Internet_Censorship_White_Paper.pdf, 2007.

[15] Dingledine, R., and Mathewson, N. Design of a Blocking-Resistant Anonymity System. https://svn.torproject.org/svn/projects/design-paper/blocking.html.

[16] Dingledine, R., Mathewson, N., and Syverson, P. Tor: The Second-Generation Onion Router. In *USENIX Security Symposium* (2004).

[17] Electron Framework. https://electronjs.org/.

[18] Fifield, D., Hardison, N., Ellithorpe, J., Stark, E., Boneh, D., Dingledine, R., and Porras, P. Evading censorship with browser-based proxies. In *Privacy Enhancing Technologies* (2012), Springer, pp. 239–258.

[19] Fifield, D., Lan, C., Hynes, R., Wegmann, P., and Paxson, V. Blocking-resistant Communication through Domain Fronting. In *PETS* (2015).

[20] Where can I find an up to date list of free US proxies? https://www.quora.com/Where-can-I-find-an-up-to-date-list-of-free-US-proxies.

[21] FlashProxy. http://crypto.stanford.edu/flashproxy/.

[22] Freedom on the Net 2017. https://freedomhouse.org/sites/default/files/FOTN_2017_Final.pdf, 2017.

[23] Geddes, J., Schuchard, M., and Hopper, N. Cover Your ACKs: Pitfalls of Covert Channel Censorship Circumvention. In *CCS* (2013).

[24] China's GitHub Censorship Dilemma. http://mobile.informationweek.com/80269/show/72e30386728f45f56b343ddfd0fdb119/.

[25] GoAgent proxy. https://code.google.com/p/goagent/.

[26] Google disables "domain fronting" capability used to evade censors. https://arstechnica.com/information-technology/2018/04/google-disables-domain-fronting-capability-used-to-evade-censors/, 2018.

[27] Everyone's Guide to By-Passing Internet Censorship for Citizens Worldwide. http://www.nartv.org/mirror/circ_guide.pdf.

[28] Hahn, B., Nithyanand, R., Gill, P., and Johnson, R. Games without frontiers: Investigating video games as a covert channel. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on* (2016), IEEE, pp. 63–77.

[29] Holowczak, J., and Houmansadr, A. CacheBrowser: Bypassing Chinese Censorship without Proxies Using Cached Content. In *The 22nd ACM Conference on Computer and Communications Security (CCS)* (2015).

[30] Houmansadr, A., Brubaker, C., and Shmatikov, V. The Parrot Is Dead: Observing Unobservable Network Communications. In *S&P* (2013).

[31] Houmansadr, A., Nguyen, G., Caesar, M., and Borisov, N. Cirripede: Circumvention Infrastructure Using Router Redirection with Plausible Deniability. In *CCS* (2011).

[32] Houmansadr, A., Riedl, T., Borisov, N., and Singer, A. I Want My Voice to Be Heard: IP over Voice-over-IP for Unobservable Censorship Circumvention. In *NDSS* (2013).

[33] Houmansadr, A., Wong, E. L., and Shmatikov, V. No Direction Home: The True Cost of Routing Around Decoys. In *NDSS* (2014).

[34] Houmansadr, A., Zhou, W., Caesar, M., and Borisov, N. SWEET: Serving the Web by Exploiting Email Tunnels. In *PETS* (2013).

[35] How Iran Censors The Internet. http://www.popsci.com/technology/article/2013-03/how-iran-censors-internet-infographic.

[36] Iran Reportedly Blocking Encrypted Internet Traffic. http://arstechnica.com/tech-policy/2012/02/iran-reportedly-blocking-encrypted-internet-traffic.

[37] Karlin, J., Ellard, D., Jackson, A., Jones, C., Lauer, G., Mankins, D., and Strayer, W. Decoy Routing: Toward Unblockable Internet Communication. In *FOCI* (2011).

[38] Khattak, S., Elahi, T., Simon, L., Swanson, C. M., Murdoch, S. J., and Goldberg, I. SoK: Making sense of censorship resistance systems. *Proceedings on Privacy Enhancing Technologies 2016*, 4 (2016), 37–61.

[39] Lantern. https://getlantern.org/.

[40] Leberknight, C., Chiang, M., Poor, H., and Wong, F. A Taxonomy of Internet Censorship and Anti-censorship. http://www.princeton.edu/~chiangm/anticensorship.pdf, 2010.

[41] Mahdian, M. Fighting Censorship with Algorithms. In *Fun with Algorithms* (2010).

[42] Matthews, P., Mahy, R., and Rosenberg, J. Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN).

[43] McPherson, R., Houmansadr, A., and Shmatikov, V. CovertCast: Using Live Streaming to Evade Internet Censorship. In *Privacy Enhancing Technologies (PETS)* (2016).

[44] meek Pluggable Transport. https://trac.torproject.org/projects/tor/wiki/doc/meek.

[45] [tor-project] summary of meek's costs, march 2017. https://lists.torproject.org/pipermail/tor-project/2017-April/001097.html.

[46] Moghaddam, H., Li, B., Derakhshani, M., and Goldberg, I. SkypeMorph: Protocol Obfuscation for Tor Bridges. In *CCS* (2012).

[47] Nasr, M., and Houmansadr, A. Game of decoys: Optimal decoy routing through game theory. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016), ACM, pp. 1727–1738.

[48] Nasr, M., Zolfaghari, H., and Houmansadr, A. The waterfall of liberty: Decoy routing circumvention that resists routing attacks.

[49] Nobori, D., and Shinjo, Y. VPN Gate: A Volunteer-Organized Public VPN Relay System with Blocking Resistance for Bypassing Government Censorship Firewalls. In *NSDI* (2014), pp. 229–241.

[50] A Simple Obfuscating Proxy. https://www.torproject.org/projects/obfsproxy.html.en.

[51] Perta, V., Barbera, M., Tyson, G., Haddadi, H., and Mei, A. A glance through the VPN looking glass: IPv6 leakage and DNS hijacking in commercial VPN clients. *Proceedings on Privacy Enhancing Technologies 2015*, 1 (2015), 77–91.

[52] Tor: Pluggable Transports. https://www.torproject.org/docs/pluggable-transports.html.en.

[53] Psiphon. http://psiphon.ca/.

[54] Access Now and EFF Condemn the Arrest of Tor Node Operator Dmitry Bogatov in Russia. https://goo.gl/nNmP86.

[55] Schuchard, M., Geddes, J., Thompson, C., and Hopper, N. Routing around decoys. In *ACM CCS* (2012).

[56] SnowFlake Pluggable Transport. https://github.com/keroserene/snowflake.

[57] Ten Ways to Discover Tor Bridges. https://blog.torproject.org/blog/research-problems-ten-ways-discover-tor-bridges.

[58] Tor Metrics. https://metrics.torproject.org/.

[59] How Governments Have Tried to Block Tor. https://svn.torproject.org/svn/projects/presentations/slides-28c3.pdf.

[60] Tschantz, M. C., Afroz, S., Paxson, V., et al. SoK: Towards Grounding Censorship Circumvention in Empiricism. In *Security and Privacy (SP), 2016 IEEE Symposium on* (2016), IEEE, pp. 914–933.

[61] Ultrasurf. http://www.ultrareach.com.

[62] uProxy. https://www.uproxy.org/.

[63] uProxy's Shadowsocks version. https://github.com/uProxy.

[64] uProxy's WebRTC version. https://github.com/UWNetworksLab/uProxy-p2p.

[65] Vines, P., and Kohno, T. Rook: Using video games as a low-bandwidth censorship resistant communication platform. In *Proceedings of the 14th ACM Workshop on Privacy in the Electronic Society* (2015), ACM, pp. 75–84.

[66] How does your VPN speed Measure against other VPNs in China? https://cc.greatfire.org/en.

[67] Top 110 Free Proxy Sites – Best Free Proxy Servers List 2017. https://www.alltechbuzz.net/top-best-free-proxy-sites-servers-2016/.

[68] SoftEther VPN User Forum: VPN Gate servers blocked in China. http://forum.vpngate.net/viewtopic.php?f=11&t=42498.

[69] SoftEther VPN User Forum: Can Not Be Used In China. http://forum.vpngate.net/viewtopic.php?f=11&t=38298.

[70] VPNGate: VPN Server List. http://www.vpngate.net/en/.

[71] Wang, L., Dyer, K. P., Akella, A., Ristenpart, T., and Shrimpton, T. Seeing Through Network-Protocol Obfuscation. In *ACM CCS* (2015).

[72] Wang, Q., Gong, X., Nguyen, G., Houmansadr, A., and Borisov, N. CensorSpoofer: Asymmetric Communication Using IP Spoofing for Censorship-Resistant Web Browsing. In *CCS* (2012).

[73] Wang, Q., Lin, Z., Borisov, N., and Hopper, N. rBridge: User Reputation Based Tor Bridge Distribution with Privacy Preservation. In *NDSS* (2013).

[74] Weinberg, Z., Wang, J., Yegneswaran, V., Briesemeister, L., Cheung, S., Wang, F., and Boneh, D. StegoTorus: A Camouflage Proxy for the Tor Anonymity System. In *CCS* (2012).

[75] Wilde, T. Knock Knock Knockin' on Bridges' Doors. https://blog.torproject.org/blog/knock-knock-knockin-bridges-doors, 2012.

[76] Wing, D., Matthews, P., Mahy, R., and Rosenberg, J. Session traversal utilities for NAT (STUN).

[77] Winter, P., and Lindskog, S. How the Great Firewall of China Is Blocking Tor. In *FOCI* (2012).

[78] Wolff, J. VPN Usage Around the World. https://cdn2.hubspot.net/hubfs/304927/Downloads/VPN-Usage-Around-the-World-Infographic.pdf, 2017.

[79] Wolff, J. The Internet Censor's Dilemma. https://slate.com/technology/2018/03/virtual-private-networks-become-more-popular-as-countries-restrict-their-use.html, 2018.

[80] Wustrow, E., Wolchok, S., Goldberg, I., and Halderman, J. Telex: Anticensorship in the Network Infrastructure. In *USENIX Security* (2011).

[81] Zolfaghari, H., and Houmansadr, A. Practical censorship evasion leveraging content delivery networks. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016), ACM, pp. 1715–1726.

A Potential Questions

Why cannot the Great Firewall blacklist millions of IP addresses? They can (at potentially high computation overhead) but this will cause them significant collateral damage due to false positive blockings. MassBrowser IP addresses are shared IP addresses that are used by benign non-circumvention traffic like VoIP, gaming, etc.

What's the difference between MassBrowser and Tor? MassBrowser and Tor are apples and oranges! MassBrowser is a system to defeat IP blocking—very much similar to meek and other Tor pluggable transports. In fact, as we have discussed, MassBrowser can be used as a Tor pluggable transport.

Even though your survey shows high interest from volunteers to deploy MassBrowser, in practice one DMCA complaint could be enough to scare all volunteers away. A key strength of MassBrowser compared to other volunteer-run circumvention systems is that the volunteers will not relay *any* controversial traffic. Volunteers only relay traffic to totally legal destinations like News and social networks, or act as a bridge to get to Tor. Also, each volunteer decides the destinations she proxies to (see Figure 8).

Roger Dingledine always argues that anything that's weaker than Tor will have inadequate security. How do you counter this? Please refer to our discussion of the "separation of properties" principle. Most of the users in China and Iran only need blocking resistance and do not need anonymity. For users who need anonymity they can use MassBrowser as a pluggable transport to Tor. When used as a pluggable transport, MassBrowser offers similar blocking resistance features to meek at a significantly lower cost of operation.

What if the censors intercept the communication between clients and buddies by apply-

ing SSL or TLS splits between country border and domain front Operator server to intercept, modify, and drop communication between the clients and operator? No circumvention system, including Tor, will work against a censoring adversary with such capabilities.

MassBrowser's certificate installation on client seems intrusive? Absolutely not! This is a local certificate. The certificate is generated locally on the user's own machine and never leaves the machine. The user is warned never to share the certificate, and advised to revoke the certificate if he or she no longer wishes to use MassBrowser. Also, our code is undergoing a code review by a third-party.

What if the censor blocks all p2p traffic? This will likely impact many MassBrowser connections, but also a significant number of legitimate p2p traffic like VoIP, gaming, file sharing, etc.

Do all Buddies have to be NATed? No! MassBrowser will work fine for Buddies who are not behind NATs as long as they are not blocked. For Buddies with static public IP addresses, MassBrowser's resistance against blocking is similar to other proxy-based circumvention systems. However, MassBrowser will only assign such Buddies to Clients with sufficient reputation as described in Section 7.2.

Will there be enough bandwidth available through Buddies to make the system practical? MassBrowser's design makes it require significantly less bandwidth than a system like Tor. This is because **1)** Tor traffic must pass through three hops in the network, **2)** only a small number of the websites browed by a user will pass through the MassBrowser network, and **3)** only a small portion of a website's contents will pass through MassBrowser's network (due to MassBrowser's deployment of CacheBrowsing).

What if a nation-state (like UAE) penalizes its citizens just for using a circumvention system? Then MassBrowser will not be the right solution for the users of such countries. Fortunately, for major censoring governments, including China, Turkey, and Iran, there is almost no instance of such punishments.

B MassBrowser Code

We have fully implemented MassBrowser as an end-user software, and it is currently in the beta release state with users evaluating it. Our current implementation of MassBrowser supports Mac, Windows, and Linux operating systems, available at `https://massbrowser.cs.umass.edu/`. In the following we give details of our system implementation.

B.1 The Operator server

We have coded Operator server mostly in Python with the Django web framework [2]. We have hosted our Operator server on Amazon CloudFront CDN [13], therefore it is a domain fronted service and can not be blocked. Our Operator's API is accessible through both standard HTTP requests and WebSockets, though we refrain from using WebSocket connections for the Client in order to prevent introducing protocol fingerprints.

As previously mentioned, the Operator maintains a database of supported websites along with per-region censorship and CacheBrowsing information for all domains in the websites. To do so, the Operator has a *probing* component that regularly crawls the supported websites to identify domains and update its information.

B.2 Buddy Software

We have coded our Buddy software in Javascript ES6 using NodeJS with a graphical user interface developed with the Electron framework [17]. In addition to the GUI interface, or Buddy software is also available as a command-line application for expert volunteers. The Buddy actively maintains a WebSocket connection to the Operator, and will be notified of newly created sessions on this channel.

The Buddy software allows volunteers to have full transparency and control over their desired settings including bandwidth limits, destination whitelists and Client blacklists (Figure 8 displays a snapshot of a Buddy volunteer configuring her destination whitelists through the GUI). The Buddy software runs with minimal interference from the user. It is able to run in the background while providing an easily accessible switch for disabling the Buddy's activities on the users demand.

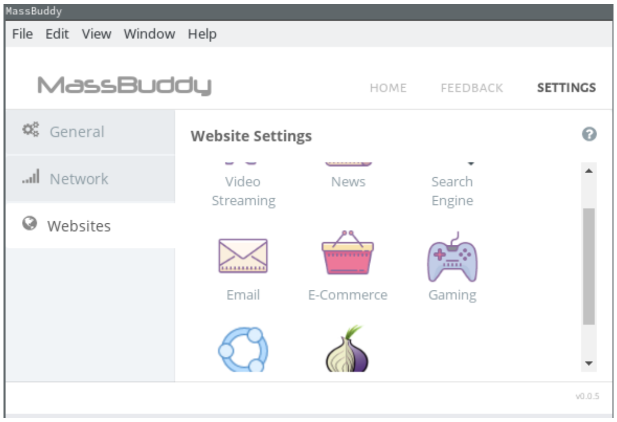
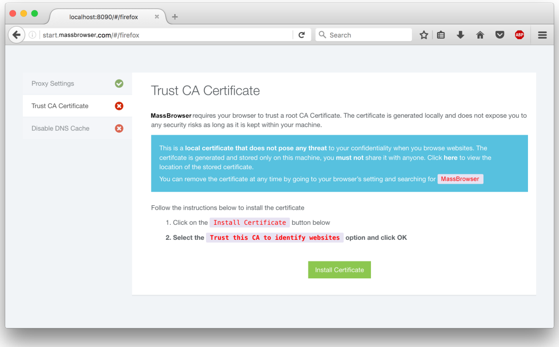


Figure 8: The settings page in the MassBrowser Buddy software allowing the user to select it's allowed content types

B.3 Client Software

We have implemented our Client software with NodeJS with an Electron based GUI. A client application, e.g., a web browser, can connect to the Client software via a SOCKS proxy. On the first run, the Client software will walk the user through a setup wizard which will assist them in configuring their preferred browsers to use MassBrowser. The current implementation of Client software provides a setup wizard for the Firefox browser only, but an expert client can set up any web browser to use the Client software. Figure 9 displays the Client setup wizard for Firefox.

The MassBrowser Client software requires to see each individual request, even when encrypted with TLS. In the normal case, the proxied TLS requests would not be visible to the Client software since it does not own the website certificates. To enable the interception of TLS connection by Client, the setup wizard adds a *locally created root certificate* to the client's browser during the initial setup. Note that the root certificate does not leave the client's computer, and therefore the client is secure as long as she does not share the certificate with others (Figure 9 shows how the user is informed during the setup). Client uses this certificate to "locally" man-in-the-middle MassBrowser's TLS connections to perform load optimizations like CacheBrowsing.



Trust CA Certificate

MassBrowser requires your browser to trust a root CA Certificate. The certificate is generated locally and does not expose you to any security risks as long as it is kept within your machine.

This is a **local certificate that does not pose any threat** to your confidentiality when you browse websites. The certifcate is generated and stored only on this machine, you **must not** share it with anyone. Click **here** to view the location of the stored certificate.

You can remove the certificate at any time by going to your browser's setting and searching for `MassBrowser`

Follow the instructions below to install the certificate

1. Click on the `Install Certificate` button below
2. **Select the `Trust this CA to identify websites` option and click OK**

Install Certificate

Figure 9: The Client setup wizard. This page is asking the user to trust the local MassBrowser root certificate and providing users with details on how to keep their connections safe.

C Complete User Survey

The following is the online survey we used in our study.

- Are you willing to voluntarily install and run Helper on your personal laptop/desktop (so you help censored Internet users)? Assume that running Helper does not cost you anything, but also does not earn you money. Also, assume that you can completely control the use of Helper (as will be asked in the follow up questions).
  - Yes
  - No
- Are you willing to install and run Helper on your personal laptop/desktop (so you help censored Internet users) if you get paid?
  - Yes
  - No

- What fraction of your unused Internet bandwidth are you willing to allocate to Helper (the unused bandwidth is the bandwidth you are not using anyways)?
  - 1%
  - 1 – 5%
  - 5 – 10%
  - 10 – 50%
  - 50 – 75%
  - 75 – 100%
- When you install the Helper software, some censored users (whom you don't know) will use your computer to connect to censored Internet websites. So your Internet provider may assume that you are browsing those websites yourself. What kind of websites do you feel comfortable (and allow) to be proxied through your computer by censored users?
  - I am OK with all websites
  - I am OK with all legal websites
  - I want to be more specific with my choices
- Which categories would you allow censored users to browse through your computer (assume that all categories use the same bandwidth)?
  - News pages (CNN, FoxNews, etc)
  - Social media (Facebook, Twitter, Instagram)
  - Search engines (Google, Bing)
  - Video sharing and streaming (YouTube, Vimeo, etc)
  - Scientific websites
- Users from which censored countries are you willing to help?
  - Any Country
  - China
  - Iran
  - Syria
  - Turkey
  - Saudi Arabia
- What is your age? (Optional)
  - 18-30
  - 30-40
  - Above 40
  - Prefer not to answer
- What is you gender? (Optional)
  - Male
  - Female
  - Prefer not to answer
- How woud you rate your computer proficiency? (Optional)
  - High
  - Medium
  - Low
  - Prefer not to answer
- Where do you live? (Optional)
  - USA
  - Europe
  - Asia
  - Other
  - Prefer not to answer

D Survey Demography

Table 4 shows the demography of our survey participants.

Table 4: The demography of survey participants

		CS	OSN	XCensored	MT	Aggregated
Gender	Male	76%	64%	73%	42%	49%
	Female	19%	14%	26%	54%	46%
	Not answered	4%	21%	0%	4%	5%
Location	USA	89%	28%	100%	100%	95%
	Europe	2%	50%	0%	0%	3%
	Asia	6%	7%	0%	0%	2%
	Other	0%	0%	0%	0%	0%
	Not answered	2%	14%	0%	0%	1%
Age	18-30	78%	57%	73%	73%	71%
	30-40	12%	28%	26%	18%	19%
	Above 40	8%	7%	0%	6%	6%
	Not answered	0%	7%	0%	1%	1%
Proficiency	High	86%	85%	53%	80%	80%
	Medium	10%	7%	46%	17%	16%
	Low	0%	0%	0%	0%	0%
	Not answered	2%	7%	0%	3%	3%