VPN Gate: A Volunteer-Organized Public VPN Relay System with Blocking Resistance for Bypassing Government Censorship Firewalls

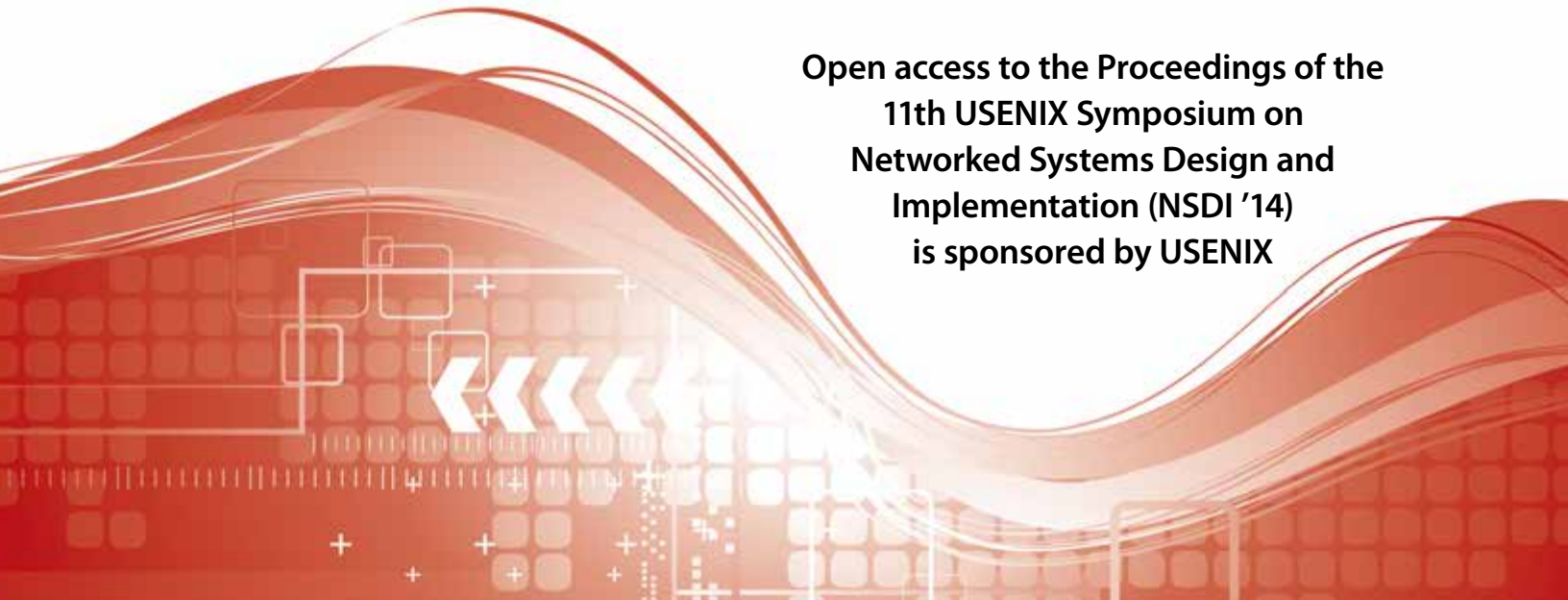
Daiyuu Nobori and Yasushi Shinjo, *University of Tsukuba*

https://www.usenix.org/conference/nsdi14/technical-sessions/presentation/nobori

This paper is included in the Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI '14).

April 2–4, 2014 • Seattle, WA, USA

ISBN 978-1-931971-09-6



VPN Gate: A Volunteer-Organized Public VPN Relay System with Blocking Resistance for Bypassing Government Censorship Firewalls

Operational Systems Track

Daiyuu Nobori and Yasushi Shinjo
Department of Computer Science, University of Tsukuba, Japan

Abstract

VPN Gate is a public VPN relay service designed to achieve blocking resistance to censorship firewalls such as the Great Firewall (GFW) of China. To achieve such resistance, we organize many volunteers to provide a VPN relay service, with many changing IP addresses. To block VPN Gate with their firewalls, censorship authorities must find the IP addresses of all the volunteers. To prevent this, we adopted two techniques to improve blocking resistance. The first technique is to mix a number of innocent IP addresses into the relay server list provided to the public. The second technique is collaborative spy detection. The volunteer servers work together to create a list of spies, meaning the computers used by censorship authorities to probe the volunteer servers. Using this list, each volunteer server ignores packets from spies. We launched VPN Gate on March 8, 2013. By the end of August it had about 3,000 daily volunteers using 6,300 unique IP addresses to facilitate 464,000 VPN connections from users worldwide, including 45,000 connections and 9,000 unique IP addresses from China. At the time VPN Gate maintained about 70% of volunteer VPN servers as unblocked by the GFW.

1. Introduction

Some countries in the world have censorship firewalls operated by their governments to prohibit access to servers in foreign countries. For instance, the Great Firewall (GFW) of China blocks access to Twitter, Facebook, and YouTube. Internet users in countries subject to censorship often use overseas public relay servers to bypass censorship firewalls. Public proxies, VPN servers, and Tor nodes [7] are popular examples of such relay servers. Usually, the IP addresses of relay servers are publically available for user convenience. A censorship authority can easily block these relays, however, by adding the IP addresses to its firewall blocking list. Moreover, the Chinese authority, in particular, scans for unlisted Tor nodes and blocks them automatically [19]. Tor relays currently have no blocking resistance [12] against such scanning activities.

In this research, we have built a public VPN relay server system with blocking resistance to censorship firewalls such as the GFW. We call this system VPN Gate. To achieve blocking resistance, VPN Gate uses frequently changing IP addresses that are provided by volunteers. The central list server, called the *VPN Gate List Server*, manages a list of the IP addresses of all active VPN servers. We call this list the *Server List*. A user can get only part of the Server List and connect his/her PC to an active VPN server in the list. The user can then communicate with blocked Internet servers through the active VPN server. It is hard for a censorship authority to block all the active VPN servers in VPN Gate.

It is important for anti-censorship systems to achieve blocking resistance. We adopted two techniques for blocking resistance: innocent IP mixing and collaborative spy detection. In innocent IP mixing, we include a number of IP addresses, which are unrelated to VPN Gate, in the Server List. For instance, we include vitally important servers (e.g., Windows Update servers). This technique forces a censorship authority to remove innocent IP addresses from the Server List before adding addresses to the firewall blocking list. The second technique, collaborative spy detection, seeks probing activities from censorship authority's computers, called *spies*. In this technique all the volunteer VPN servers work together to create a source IP address list of spies, called the *Spy List*, and they ignore probing packets from spies. This technique makes the authority unable to distinguish between the IP addresses of active VPN servers and innocent IP addresses or those of inactive VPN servers.

The VPN Gate system consists of instances of the VPN Gate Server software, an optional application, the VPN Gate Client software, and a central List Server. Volunteers can easily install and execute VPN Gate Server. For instance, volunteers don't have to configure Network Address Translation (NAT) boxes to open TCP/UDP ports. Users can connect to VPN Gate Server with a Secure Sockets Layer (SSL)-VPN protocol by using VPN Gate Client. Users can also connect to a VPN server with the L2TP/IPsec, OpenVPN, and MS-SSTP protocols by using the built-in, OS-provided VPN clients on PCs and smartphones. As for the third piece of the system, our research group runs the VPN Gate List Server which accepts registration from volunteer servers, generates the Server List, and distributes it to users.

We launched VPN Gate on March 8, 2013. On August 29, we had about 3,000 active VPN Gate servers. This number is comparable to the number of Tor relay nodes. On the same day we had 464,000 connections to the VPN Gate servers. These connections were from 88,000 unique source IP addresses.

VPN Gate has blocking resistance against the GFW. Shortly after we started the service, the GFW authority added the IP addresses of all the volunteer servers into the GFW blocking list. On April 4, the GFW blocked 81% of all volunteers, so only 19% of active volunteers were reachable from China. Hence, we implemented the innocent IP mixing and collaborative spy detection techniques. As a result, we achieved 50% reachability from China on April 26, and 75% on May 9. Moreover, around 40% of our volunteers' IP addresses changed every day. The GFW could not catch up to our increasing number of volunteers and their changing IP addresses. VPN Gate has thus provided stable reachability for Chinese users. At the end of August, 2013, we have about 45,000 daily connections from 9,000 unique IP addresses in China, while Tor had an estimated 3,000 users from China.

VPN Gate is a system for bypassing censorship. It is not an anonymizer. Unlike Tor, VPN Gate volunteer servers record packet logs. VPN Gate also has no multi-hop relaying function.

2. Related Work

VPN Gate organizes VPN servers provided by volunteers. This method is similar to that of the well-known anonymizer Tor [7]. Since communications in Tor are relayed by three Tor nodes to achieve anonymity, they are slow.

Tor nodes are classified into two types: public *relays* and non-public *bridges*. It is easy for censorship authorities to block the public relays. Users behind censorship firewalls must find non-public bridges through web sites, email, and other means of contact. Although bridges are not public, censorship authorities can probe and block them [18, 19]. Using obfsproxy, it is possible to obfuscate the network traffic exchanged between Tor clients and bridges [17]. However, Tor bridges currently have no blocking resistance against such probing activities.

Unlike Tor, VPN Gate focuses on bypassing censorship firewalls and does not provide anonymity. Since communications in VPN Gate are relayed by a single VPN server, they are much faster than in Tor. To use VPN Gate, users behind censorship firewalls must get a list of VPN servers through web sites, email, and so forth. Unlike Tor, VPN Gate also includes innocent IP addresses in a list of VPN servers. We describe this aspect in Section 4.2. Furthermore, VPN Gate has a mechanism making it harder for censorship authorities to probe VPN servers. We describe this aspect in Section 4.3.

It is not trivial to run Tor relay and bridge nodes. Rbox-Tor helps volunteers run Tor nodes by using virtual machines [16]. VPN Gate also helps volunteers run VPN servers by a variety of techniques, including Network Address Translation (NAT) traversal capability. We describe this capability in Section 5.2.

VPN Gate maintains the list of VPN servers in a centralized server. This mechanism is similar to a *tracker* in BitTorrent [6]. It is easy for censorship authorities to block communications to a tracker. To avoid using centralized trackers, BitTorrent introduced a distributed hash table (DHT), implemented in the Mainline and Azureus DHTs [2, 4]. We chose a centralized server instead of a DHT for two reasons. First, we have to return a different partial server list for each client. Second, we have to accumulate all information from all active VPN servers in a central server to analyze unusual usages. We describe these design choices in Section 4.3.

Many researchers are working on censorship-resistant systems [3, 5, 8, 9, 13, 14, 20]. These systems either are Web-access-specific ones or require modifying existing protocol stacks. Here, in contrast, we describe a VPN-based censorship-resistant system that allows using arbitrary protocols without modifying an existing protocol stack.

Many free and commercial VPN services are also used to bypass censorship firewalls [10]. Since most such services use a set of centralized VPN servers with fixed IP addresses, censorship authorities can easily block these services with firewalls. Some VPN services do have a decentralized or peer-to-peer (P2P) architecture [11, 15]. There have been no published reports or results, however, on bypassing methods.

Finally, our collaborative spy detection technique is similar to collaborative intrusion detection [22]. In this paper, we show a specific method to protect VPN servers.

3. VPN Gate Overview

Figure 1 shows an overview of VPN Gate. A volunteer downloads the VPN Gate Server software and runs it on a PC. While VPN Gate Server is running, it registers itself to the VPN Gate List Server. This server maintains the Server List, a list of IP addresses for active VPN Gate Server instances.

Assume here that a VPN Gate user lives behind a censorship firewall and cannot access blocked servers in foreign countries. The user first accesses a web page for the VPN Gate List Server to get a list of VPN servers. To avoid discovery of all VPN servers by censorship authorities, VPN Gate List Server returns only a small part of the entire Server List. Next, the user chooses a VPN Gate Server instance from the partial list. Finally, the user connects his/her PC to the chosen server by using either a native VPN client on the PC or dedicated VPN client

software, called VPN Gate Client. Once the VPN connection is established, the VPN server relays all the user's communications to the Internet.

3.1. Hosting VPN Gate Server as a volunteer

As described above, a volunteer installs and runs VPN Gate Server on a PC. At this time, the volunteer does not need to show his/her name, address, or any other personal information. While VPN Gate Server is running, it waits for new VPN connections from users. It accepts four VPN protocols: L2TP/IPsec, OpenVPN, MS-SSTP, and SoftEther VPN Protocol.

While VPN Gate Server is running, it periodically checks the type of Internet connection on the PC. If the PC is behind a NAT box, VPN Gate Server attempts to open a port via Universal Plug and Play (UPnP) or UDP hole punching. With the recognized type of Internet connection, VPN Gate Server registers itself to VPN Gate List Server, which we describe in Section 3.3.

3.2. Connecting to VPN Gate as a user

A VPN Gate user accesses the web site of the VPN Gate List Server and obtains part of the Server List. This contains information about volunteer servers, including IP addresses and port numbers, geographic locations, line quality parameters such as bandwidth and delay, numbers of current VPN connections, and numbers of cumulative VPN connections. The user thus chooses a preferred VPN server from the subset of the Server List.

Since censorship authorities can easily discover the web site of the VPN Gate List Server, a user in a country subject to censorship likely cannot access the web site directly. Such a user can instead access it via an HTTP relay site provided by VPN Gate Server. Section 4.5 gives the details of this mechanism.

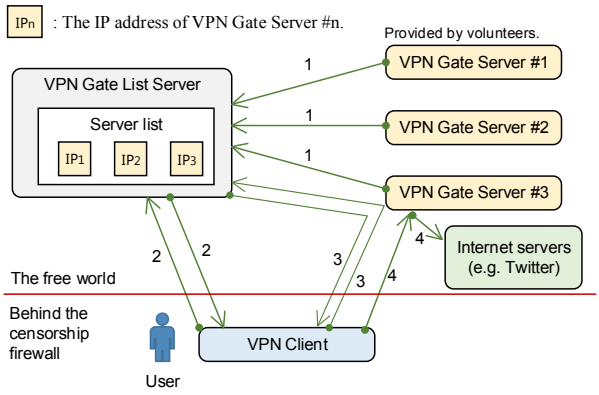
Next, the user can establish a VPN connection by using one of the following methods:

1. Using a built-in VPN client in the operating system (OS).

The user inputs the IP address of the chosen VPN server in the configuration window of the L2TP/IPsec or MS-SSTP VPN client. In this window, the user also fills in the user name and password fields with fixed values, "vpn" and "vpn". The advantage of this method is that it does not require installing any software.

2. Using OpenVPN Client.

The user installs the OpenVPN Client software once. Then, he/she downloads an OpenVPN connection setting file (.ovpn file) from the VPN Gate List Server web site and runs OpenVPN Client with the same setting file each time when he/she connects to VPN Gate.



1. Register itself to the VPN Gate List Server.
2. Get the server list directly.
3. Get the server list with the Indirect Server List Transfer Protocol.
4. Access to Internet servers through the VPN server.

Figure 1. Overview of VPN Gate.

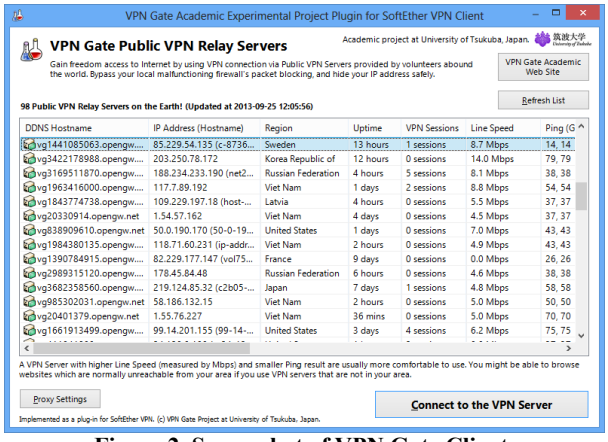


Figure 2. Screenshot of VPN Gate Client.

3. Using VPN Gate Client.

The user installs the VPN Gate Client software once and runs it each time he/she needs a VPN connection. VPN Gate Client displays the user's portion of the Server List, as shown in Figure 2, and he/she chooses a server for connection. The advantage of this method is that it is easy, and it supports the Indirect Server List Transfer Protocol, which we describe in Section 4.4.

3.3. VPN Gate List Server

The VPN Gate List Server software accepts registrations from active VPN Gate Server instances and monitors these servers' statuses. When VPN Gate List Server receives a server list request from a client, it returns a small part of the Server List. In addition, VPN Gate List Server implements the firewall resistance system described in Section 4.

4. Firewall Resistance System

The firewall resistance system in VPN Gate achieves blocking resistance to censorship firewalls. This system is implemented in both VPN Gate Server and VPN Gate List Server. In this section, we first briefly describe the blocking methods of the Chinese censorship firewall. After that, we describe our blocking resistance techniques. The two key techniques are *innocent IP mixing* and *collaborative spy detection*.

4.1. Blocking methods used in the Great Firewall of China

We set our goal in designing the system to achieve blocking resistance to the Chinese GFW. To do so, we studied the GFW's blocking methods. According to various reports [1, 5, 21], the GFW exists at borders between Chinese internet service providers (ISPs) and overseas ISPs, and it can block all IP packets sent to IP addresses on the blocking list. The GFW authority must maintain a blocking list of IP addresses. It exploits both human resources and automated scanners to maintain the blocking list. For instance, the GFW authority performs scanning to detect hidden Tor nodes [19].

4.2. Innocent IP mixing

The first technique for achieving blocking resistance in VPN Gate is innocent IP mixing, illustrated in Figure 3. In this technique, we include a number of fake IP addresses, called *innocent IP addresses*, when VPN Gate List Server returns a list of VPN servers to a user. Innocent IP addresses are chosen from among addresses unrelated to VPN Gate, and they should be addresses of vitally important hosts in the Internet. Examples of good innocent IP addresses include DNS root servers, top-level-domain DNS servers, Windows Update servers, and popular email servers. After a censorship authority notices innocent IP mixing, it cannot automatically add all obtained IP addresses from the Server List to its firewall blocking list. Instead, the authority has to verify whether each of the obtained IP addresses is the real IP address of a VPN Gate Server. We do not have to mix innocent IP addresses every day, all the time; it is sufficient to mix in a small number of innocent IP addresses occasionally to keep the authority's attention.

As a disclaimer, we have included the following warning sentence on the web site for the VPN Gate List Server: "This server list might contain wrong IP addresses, and authorities should not use these IP addresses for firewall blocking lists."

Innocent IP mixing does not affect regular users of VPN Gate. If a user occasionally chooses an innocent IP address, he/she will just get a connection error. The user can then simply try another IP address from the Server List.

Innocent IP mixing does not cause distributed denial of service (DDoS) attacks on innocent servers. Suppose that we have 100 million users each day, and we mix in one innocent IP address for every 1,000 real VPN servers. If each user chooses a target VPN server randomly from the list, the server for an innocent IP address will receive an expected 100,000,000 / 1,000 = 100,000 connection requests each day. If we assume five retry packets per connection request, the server will receive 7 useless packets per second. We believe that such a small number of useless packets is harmless to Internet servers of the present day.

In practice, a typical user does not choose a VPN server randomly but tries servers from top to bottom in the list. A user's list typically has 100 VPN servers, and we can put the innocent IP address in the middle of the list. Since the user will most likely succeed in connecting or stop trying before reaching the innocent IP address, the corresponding server will never receive any connection requests.

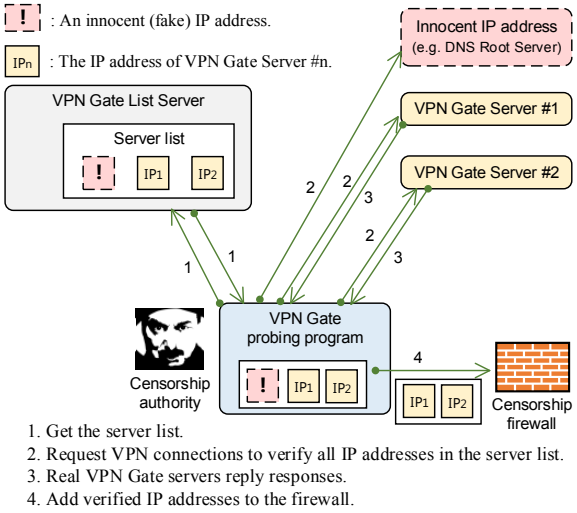


Figure 3. Innocent IP mixing.

4.3. Collaborative spy detection

The second technique for achieving blocking resistance in VPN Gate is collaborative spy detection. This technique detects probing activities from the computers of a censorship authority, called *spies*. To find spies, all instances of VPN Gate Server work together and build a source IP address list of spies, called a *Spy List*. As illustrated in Figure 4, the servers then ignore probing packets from spies in the Spy List. The Spy List contains both IP addresses and ranges of IP addresses. This technique prevents censorship authorities from distinguishing whether VPN Gate Server is running on a specific IP address.

Collaboration is vital to detecting spies in VPN Gate. Since a spy establishes a VPN connection with regular VPN protocol procedures, a single VPN Gate Server instance cannot distinguish between a spy and a regular client. When a single VPN Gate Server instance does find a spy by recognizing the unusual behavior of the spy client, it is too late because the spy has already succeeded in discovering the VPN server by that time. Therefore, the VPN server must distinguish whether a client is a spy before sending its first response to the client. This is impossible for a single VPN Gate Server instance.

To solve this problem, all VPN Gate Servers work together to detect spies, share the Spy List, and deny connections from clients in the Spy List. The process of generating the Spy List consists of the following two procedures:

1. Procedure in VPN Gate Server

VPN Gate Server records VPN connection logs, which we classify into three types: *complete calls*, *incomplete calls*, and *tiny calls*. A complete call means a VPN connection that is normally established between a client and a server, where the amount of actual data transfer exceeds a threshold. An incomplete call is a VPN connection that is disconnected either by a client before a negotiation completes or because of a protocol error. A tiny call is a VPN connection that has either a very short duration or a small amount of data transfer. VPN Gate Server records all these calls with metadata such as source IP addresses, times, data transfer amounts, and durations. Each VPN Gate Server instance regularly sends these logs to the VPN Gate List Server.

2. Procedure in VPN Gate List Server

VPN Gate List Server aggregates the logs from all VPN Gate Server instances in order to find spies by using the following conditions:

I. If many VPN servers received incomplete calls from a specific IP address or a specific range of IP addresses, we mark the address or range as a spy.

II. If many VPN servers received tiny calls from a specific IP address or a specific range of IP addresses, we mark the address or range as a spy.

VPN Gate List Server performs this procedure periodically and distributes the generated Spy List to all VPN servers. We reduce the size of the Spy List by aggregating multiple IP addresses into a range of IP addresses in a /24 block. We apply this aggregation technique when the number of IP addresses in a block exceeds a threshold, which varies according to the frequency of accesses and other conditions. For example, the threshold for Chinese IP addresses is smaller than that for United States IP addresses.

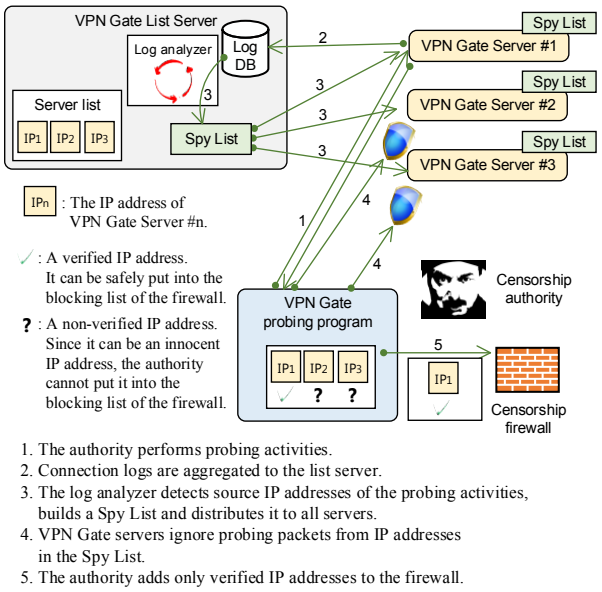


Figure 4. Collaborative spy detection.

4.4. Distributing server lists to users

Anti-censorship systems that use relay servers face the relay server discovery problem: how can clients discover relay servers without having a censorship authority also discover and block these servers [9]? To address this problem, VPN Gate applies several techniques.

First, we use a technique called *keyspace hopping* [9]. In keyspace hopping, each client pseudorandomly uses a unique set of servers, just as a wireless node uses frequency hopping to resist jamming. This technique ensures that each client discovers only a small fraction of the total number of VPN Gate servers. Furthermore, we use the network address of a client as the seed of a pseudorandom number generator in keyspace hopping. This method forces the censorship authority to have a large number of IP addresses in order to collect the IP addresses of all the VPN Gate servers.

The second technique is to introduce the *Indirect Server List Transfer Protocol*. When a user in a country subject to censorship tries to get a fresh server list through VPN Gate Client, a firewall will likely block the communication with VPN Gate List Server. We thus implemented the Indirect Server List Transfer Protocol to solve this problem. This protocol allows VPN Gate Client to get a fresh server list via an intermediate server. The intermediate server is a VPN Gate Server instance known by the client. Note that a server list transferred with this protocol is digitally signed to prevent modification by the intermediate server.

The third technique is dynamic generation of initial server lists. It is useful for a first-time user of VPN Gate Client to have a fresh initial list of VPN Gate servers. To

achieve this, our download Web server dynamically generates a fresh initial server list for each destination and includes it in the installation package of VPN Gate Client. We generate this initial list by applying keyspace hopping, the first technique mentioned above. We also mix innocent IP addresses into the initial list.

On August 19, 2013, our VPN Gate List Server accepted about 379,000 indirect server list transfer requests, representing 23.2% of the total of about 1,630,000 user requests on that date.

4.5. HTTP relay function and Daily Mirror URL Mail Service

It is easy for a censorship authority to block our download web server and the web site of the VPN Gate List Server. To overcome this problem, we implement an HTTP relay function in VPN Gate Server. This function gives users the chance to download VPN Gate Client at the time of first use. This function also provides access to the VPN Gate Server List web site for those who use built-in VPN clients.

As we described in Section 4.1, censorship firewalls can detect and block our HTTP relay function by keyword inspection. To make this inspection task difficult, we respond with gzip-compressed HTTP contents.

VPN Gate also provides a Daily Mirror URL Mail Subscription service. This service emails the latest URL list to subscribers every day. Each list contains the URLs of a small number of active VPN Gate Server instances that enables the HTTP relay function. These URLs are suitable for distribution via online and offline social networks in countries subject to censorship. On September 13, 2013, we had 11,000 subscribers to this mail service. Through keyspace hopping, we disclose only a small fraction of VPN Gate servers in this service. When a subscriber signs up the service, we record the IP address of the subscriber's Web browser and use it as the seed of a pseudorandom number generator for keyspace hopping.

5. Implementation

In this section, we describe the implementation of VPN Gate Server, VPN Gate Client, and VPN Gate List Server.

5.1. Implementation of VPN Gate Server

We have implemented VPN Gate Server as an application program for Windows. The program code is based on SoftEther VPN Server, which is our free VPN server program[1]. VPN Gate Server supports the following four VPN protocols. VPN Gate Server treats all VPN clients using any of these VPN protocols equally.

1. L2TP/IPsec
2. OpenVPN protocol
3. MS-SSTP
4. SoftEther VPN protocol

The SoftEther VPN protocol implements Ethernet over SSL on TCP or UDP. It has affinity with most firewalls. It requires VPN Gate users to install the specific VPN Gate Client in their devices. Unlike MS-SSTP, this VPN protocol is usable in UDP-only environments.

We have also implemented an *Internet sharing function* in VPN Gate Server. This function allows sharing of a single outgoing IP address for the server while allocating a different private address for each VPN client.

5.2. Running VPN Gate Server behind a NAT box

We assume that the PCs of most volunteers running VPN Gate Server are behind NAT boxes. To increase the number of available volunteer servers, it is necessary to make VPN servers reachable from the Internet even when they are behind NAT boxes. Therefore we implemented an automatic port-opening function in VPN Gate Server, via UPnP and UDP hole punching. This function also works in the intermediate servers for the Indirect Server List Transfer Protocol described in Section 4.4.

To increase NAT affinity, we also added UDP support to our SoftEther VPN protocol. The previous SoftEther VPN Protocol was based on SSL and worked only with TCP. To extend it to work with UDP, as well, we designed and implemented a Reliable UDP (RUDP) protocol that has a retransmission control mechanism like that of TCP.

5.3. Status monitoring of VPN servers

VPN Gate List Server performs status checking of all registered VPN servers. It executes this checking not only the first time it registers a VPN server but also periodically thereafter. After VPN Gate List Server verifies that a VPN server is functional, it adds an entry for the VPN server into the Server List.

In addition to functional checking, VPN Gate List Server collects the Internet connection qualities of registered VPN servers. To measure communication delays of the last one mile network, each VPN server sends ICMP echo requests to the Google Public DNS server (8.8.8.8)[2]. To measure communication bandwidths, each VPN server runs a TCP speed test tool with our speed test servers. The VPN servers then report these results to

[1] http://www.softether.org/

[2] Google Public DNS server is located around the world. https://developers.google.com/speed/public-dns/faq

the VPN Gate List Server. Users can view these results on the List Server's web site, thus enabling them to choose a good VPN server instance with a low-delay, high-bandwidth Internet connection.

5.4. VPN connection logs and packet logs

Each VPN server records VPN connection logs when a VPN client establishes a tunnel, and when the user disconnects the tunnel. Each VPN server also records packet logs that include not only TCP and UDP headers but also payloads. A volunteer can read these logs and know the source IP addresses of VPN clients. When a criminal uses a VPN server, the owner of the server may pass these logs to a public authority. The VPN servers also transmit the VPN connection logs to the VPN Gate List Server, which uses them for collaborative spy detection, as described in Section 4.3.

Each VPN server records VPN connection logs when a VPN client establishes a tunnel, and when the user disconnects the tunnel. The VPN servers transmit the VPN connection logs to the VPN Gate List Server, which uses them for collaborative spy detection, as described in Section 4.3.

In addition to connection logs, each VPN server also records the following minimum information for packet logging.

1. TCP Packets

The VPN server records the IP and TCP headers of SYN, SYN+ACK, and ACK packets. It records no payloads other than HTTP request headers.

2. UDP Packets

The VPN server records the IP and UDP headers for DHCP and IPsec/UDP and OpenVPN/UDP initiate packets. We record the headers because these VPN protocols can be used to hide client IP addresses. The VPN server does not record payloads.

Since we do not want VPN Gate to be used as an anonymizer, we intentionally designed it to record these logs so as to prevent abuse by criminals while maintaining the privacy of normal users. Without a packets logging function, criminals could abuse VPN Gate to hide their client IP addresses. When a criminal uses a VPN server, the owner can pass packets logs to a law enforcement agency. The "VPN Gate Anti-Abuse Policy" on the web site clearly states that each VPN Gate server records packet logs in order to prevent such abuse.

5.5. Implementation of VPN Gate Client

We implemented VPN Gate Client as an extension of SoftEther VPN Client, a VPN client program for establishing VPN connections to SoftEther VPN Server instances. SoftEther VPN Client consists of a virtual network adapter kernel-mode driver, a VPN processing module, and a GUI. We modified the GUI by adding a window to show a list of VPN servers (Figure 2). We also implemented a client module for the Indirect Server List Transfer Protocol.

Furthermore, we include the VPN Gate Server function in the VPN Gate Client program. We took this idea from P2P file sharing applications such as BitTorrent. In P2P file sharing, each client also contributes to the network as a server. The VPN Gate Server function in VPN Gate Client is disabled by default. A user who wants to be a volunteer can enable this function manually. This function is also automatically disabled while using VPN Gate Client to connect to another VPN server.

5.6. Dynamic generation of VPN Gate Client package

VPN Gate users download VPN Gate Client from the download server or relays, as described in Sections 4.4 and 4.5. Every time the download server responds to a user, it generates a new ZIP package. The fixed content of each ZIP package consists of the binary of VPN Gate Client. The package also includes variable content in the form of an initial server list. The ZIP file also includes a file with a random filename and random data at the head, for blocking resistance to censorship firewalls. This technique eliminates characteristics of TCP streams for VPN Gate Client downloading traffic.

If the download server generated a temporary ZIP file each time it received a request, this would increase the disk I/O load at the server. To eliminate this load, we implemented a lightweight, in-memory ZIP generator in the download server. For each downloading user, the ZIP generator consumes only a fixed small amount of a buffer in the server's memory.

6. Experiences

We initiated the VPN Gate web site and released the programs VPN Gate Server and Client on March 8, 2013. In this section, we demonstrate achievement of our purposes described in Section 1, by evaluating our experiences for six months after the release of VPN Gate.

6.1. Statistics of users and volunteers

Figure 5 shows the variation in the number of daily VPN connections, and Figure 6 shows the number of unique IP addresses for VPN clients on a daily basis. For instance, we had about 464,000 connections from 88,000 unique IP addresses on August 29. On this day, there

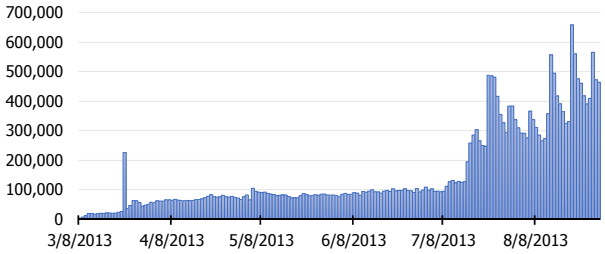


Figure 5. Number of daily VPN connections.

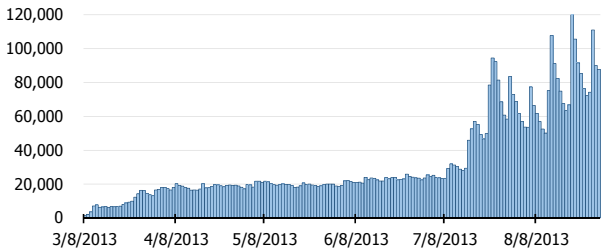


Figure 6. Number of daily unique IP addresses for VPN clients.

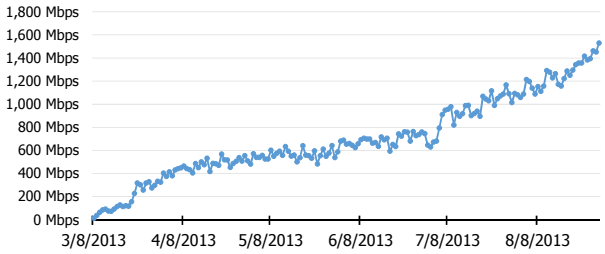


Figure 7. Daily total bandwidth for VPN connections.

were 5.3 VPN connections on average from a unique source IP address. Figure 7 shows the variation in the total bandwidth. We had a total of 1.6 Gbps on August 29. The total bandwidth steadily increased because the number of active volunteers increased along with the number of users.

Table 1 lists the top ten countries in terms of the number of client VPN connections on August 30, 2013. Table 2 lists the top ten countries in terms of the amount of client VPN traffic through August 30, 2013. Each of these tables includes China, Thailand, and Iran, countries that have censorship firewalls. VPN Gate thus helped users in these countries to bypass the firewalls. Table 2 also includes Korea, the United States, Japan, and Taiwan. Since high-speed home Internet lines are popular in Korea, it ranked first in total data transfer but only seventh in the number of VPN connections. On the other hand, while Thailand and Iran had large numbers of VPN connections, they did not yield large amounts of transferred data. This means that most users in these countries have low-speed Internet lines. These results show that the bottlenecks are mostly on the client side, and not on the server side.

Table 1. Numbers of VPN connections at different client locations.

Ranking	Location	Number of VPN connections	Percentage
1	Taiwan	7,253,003	27%
2	China	3,974,954	15%
3	Thailand	3,841,947	14%
4	Iran	2,281,446	8%
5	Japan	1,768,716	6%
6	Vietnam	1,399,833	5%
7	Korea	1,373,906	5%
8	Indonesia	742,640	3%
9	United States	589,148	2%
10	Hong Kong	466,265	2%
	190 other locations	3,536,359	13%
	Total	27,228,217	100%

Table 2. Total transferred data at different client locations.

Ranking	Location	Transferred data amount	Percentage
1	Korea	460.0 TB	35%
2	China	193.4 TB	15%
3	United States	145.7 TB	11%
4	Japan	111.1 TB	8%
5	Taiwan	90.4 TB	7%
6	Iran	45.1 TB	3%
7	Hong Kong	28.2 TB	2%
8	Malaysia	26.3 TB	2%
9	Vietnam	25.8 TB	2%
10	France	18.0 TB	1%
	190 other locations	187.1 TB	14%
	Total	1,331.1 TB	100%

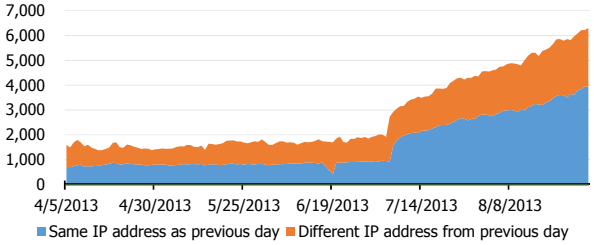


Figure 8. Numbers of VPN servers with changed and unchanged IP addresses.

We gained a total of 16,523 volunteer servers from 127 countries or regions over the course of 175 days. These servers have used 108,633 unique IP addresses.

Figure 8 shows the numbers of VPN servers with changed and unchanged IP addresses from April 5 to August 30. The lower blue area corresponds to servers whose IP addresses were unchanged from the previous day, while the upper orange area corresponds to servers whose IP addresses did change. For instance, on August 30 we had 3,935 unchanged IP addresses and 2,363 changed IP addresses. This means that 38% of the VPN servers had a different IP address from that of the previous day. On average, 40% of VPN servers had new IP addresses every day. This changing of IP addresses contributed to increasing the reachability from countries subject to censorship.

Table 3. Locations of VPN Gate Server instances on August 30, 2013.

Ranking	Location	Number of volunteer servers	Percentage
1	Korea	841	30%
2	Japan	637	23%
3	Vietnam	444	16%
4	United States	181	6%
5	Russia	119	4%
6	France	57	2%
7	Thailand	51	2%
8	United Kingdom	41	1%
9	Indonesia	38	1%
10	Canada	29	1%
	66 other locations	362	13%
	Total	2,800	100%

Table 4. Active VPN servers on August 30, 2013.

	Volunteers	Percentage
Direct connection (non-NAT)	3,884	27%
NAT (UPnP compatible)	7,384	52%
NAT (UDP hole punching compatible)	3,006	21%
Total	14,274	100%

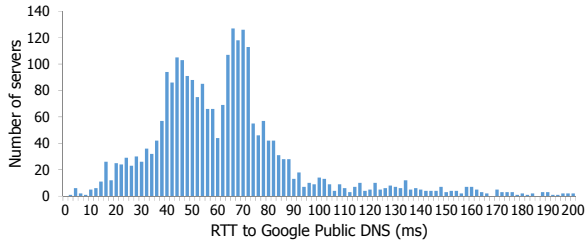


Figure 9. Round-trip time between volunteer servers and Google Public DNS on August 30, 2013.

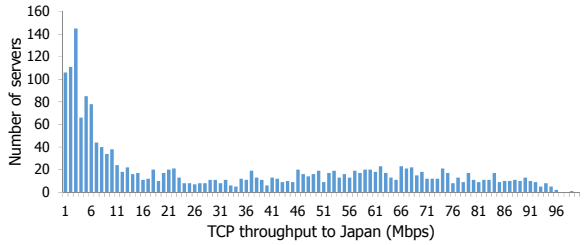


Figure 10. TCP throughput to the Japan server.

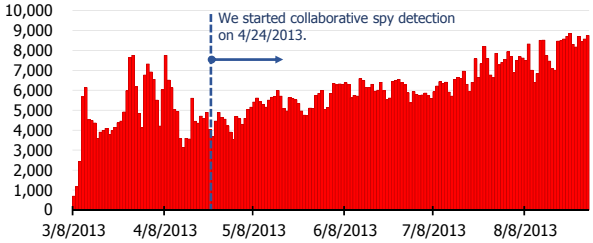


Figure 11. Number of daily unique IP addresses for VPN clients in China.

Table 3 shows the geographical distribution of the 2,800 volunteer servers running at 15:00 (GMT) on August 30, 2013. We resolved the location of each volunteer by using IP address allocation information. We found that 77% of the volunteers were from five countries: Korea, Japan, Vietnam, United States, and Russia.

We examined the quality of the Internet connections provided by the VPN servers. Figure 9 shows the round-trip times (RTTs) between each VPN server and the Google Public DNS server (IP address: 8.8.8.8), on August 30, 2013. Since Google Public DNS Servers are located worldwide, the RTT implies the quality of the last-mile line to the ISP of each VPN server. Most of the VPN servers had RTT values of 100ms or less. This means that most of the VPN servers are connected to the Internet with pretty good lines. Figure 10 shows the TCP bandwidth between each VPN server and our speed test server in Japan. More than 50% of the VPN servers had bandwidth of 5Mbps or faster. We estimated the total available bandwidth as 70Gbps. This is much larger than the used bandwidth of 1.6Gbps shown in Figure 7.

Table 4 lists the types of Internet connections used by the VPN servers on August 30, 2013. The data shows that 72.8% of VPN servers were behind NAT boxes. This means that the NAT affinity function described in Section 5.2 worked well.

6.2. Users from China

The Chinese GFW authority began to block the IP address of the VPN Gate List Server on March 12, 2013. It also began attempting to block all IP addresses of VPN servers listed on the web site for the List Server. Despite this, Figure 11 shows that the number of users from China increased continuously. This figure does not include spies detected by our collaborative spy detection. On August 29, the number of unique IP addresses for clients in China was 8,000, and this occupied 10% of all unique source IP addresses for VPN connections. These results show that our blocking resistance techniques are working effectively.

We measured the blocking rate of our VPN servers by the GFW since March 22. Early on, as shown in Figure 12, the GFW authority succeeded in blocking our VPN servers effectively. At that time only 30% of VPN servers were reachable from China. After we started innocent IP mixing and collaborative spy detection, however, the blocking rate decreased. On June 19, 78.5% of VPN servers were reachable from China. At the end of August, we typically had 60-70% of servers reachable from China. On August 8, the rate of servers blocked by the GFW suddenly decreased. We suppose that this was due to technical problems in the GFW.

In summary, we have achieved strong blocking resistance to China's GFW with VPN Gate Server. This

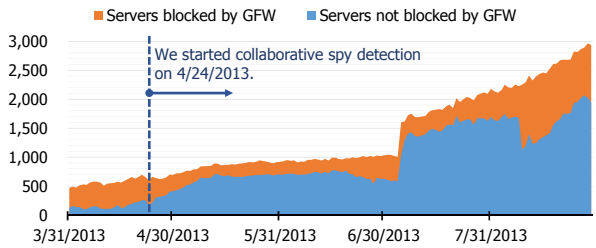


Figure 12. Numbers of VPN servers blocked and not blocked by the GFW.

was achieved by applying two techniques, namely, innocent IP mixing and collaborative spy detection. Rapid changing of server IP addresses has also contributed to this result.

6.3. Cat-and-mouse game with the Great Firewall authority

We played a cat-and-mouse game with the Chinese GFW authority until we implemented innocent IP mixing and collaborative spy detection. Here, we outline the history of this game.

March 8: We launched VPN Gate.
We initiated the web site and released the server and client programs on a Friday. Many Chinese users found VPN Gate within the first four days before blocking. On March 11, we had 5,663 unique IP addresses for clients from China. We assumed that the officers of the GFW authority did nothing on Saturday and Sunday.

March 11: GFW blocked VPN Gate List Server.
The GFW authority blocked the IP address of the VPN Gate List Server. Users in China could no longer visit the VPN Gate web site or download VPN Gate Client after this time. Some users in China began to spread URLs for the relay sites described in Section 4.5 by using domestic Chinese SNS web sites (e.g., Weibo). The relay sites helped Chinese users to visit the VPN Gate List Server web site and download VPN Gate Client. VPN Gate Client users could continue to use it with the support of the Indirect Server List Transfer Protocol.

March 12: GFW started automatic blocking.
The authority started to get the list of active VPN servers from the VPN Gate List Server periodically, and it started adding *all* IP addresses in the list to the GFW. On March 12 and 13, the authority performed this task twice a day. After March 14, the authority performed this task several times a day. We assume that the authority implemented an automated tool for this task. This response revealed that the GFW authority can discover an anti-firewall service and develop a blocking tool for it within only four days after the service starts.

March 13: We discovered a single spy IP address.
We set up 32 servers that did not run VPN Gate Server. We also added code in VPN Gate List Server to mix different portions of the IP addresses of these servers according to each request. We used these IP addresses as steganographic codes. For example, if the source IP address was 1.2.3.4, we mixed IP addresses #7, #14, #20, #21, and #27 into the list sent to the requester. Approximately 30 minutes later, the GFW blocked some of the steganographic IP addresses. We could then calculate that the IP address of the spy was 210.72.128.200. According to Whois, this is an IP address operated by China Science and Technology Network (CSTNET), an institution of the Chinese Academy of Sciences. We confirmed that the authority used this IP address to get our VPN server list and blocked this address from accessing the VPN Gate List Server web site. We also found that the user agent value of the spy program was "Python-urllib". We thus assumed that the authority wrote the spy program in Python.

March 14: GFW started getting the VPN server list from oversea cloud servers.
After we had blocked the source IP address of the authority, it started using Amazon EC2 and Gorilla Servers to get VPN server lists. We found that the user agent value was still "Python-urllib", as shown in Figure 13. The authority obtained the VPN server lists at fixed intervals. We could thus distinguish spies from regular users, but since it was easy for the authority to vary the user agent value and interval, we decided not to use these characteristics for detecting spies. The authority obtained many IP addresses of our VPN servers and put them in the GFW blocking list. After this automated process began, approximately 80% of all VPN servers became unreachable from China.

March 14: We started innocent IP mixing.
We began to mix unrelated IP addresses at the University of Tsukuba into the VPN server lists. We observed that these IP addresses became unreachable from China within 30 minutes. We tested this several times for around four hours. The GFW always blocked our newly mixed IP addresses within 30 minutes or less. This means that the GFW authority trusted our VPN server list at that time, and they did not verify the IP addresses in the list before blocking them. In other words, we had power to control the GFW for a short time.

March 16: GFW suspended using the automated tool.
The authority noticed that the VPN server lists included unrelated IP addresses. It thus suspended using the automated tool for inserting IP addresses in the VPN server

ID	Access Date	Client FQDN	URL	User Agent
3312453	3/23/13 7:40 PM	ec2-23-20-4-19.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3312674	3/23/13 7:41 PM	ec2-50-16-163-135.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3313273	3/23/13 7:45 PM	198-136-27-242.static.gorillaservers.com	http://www.vpngate.net/	Python-urllib/1.17
3313385	3/23/13 7:45 PM	ec2-23-20-4-19.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3313579	3/23/13 7:46 PM	ec2-50-16-163-135.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3314469	3/23/13 7:50 PM	ec2-23-20-4-19.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3314708	3/23/13 7:51 PM	ec2-50-16-163-135.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3315395	3/23/13 7:55 PM	ec2-23-20-4-19.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3315642	3/23/13 7:56 PM	ec2-50-16-163-135.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3316250	3/23/13 8:00 PM	198-136-27-242.static.gorillaservers.com	http://www.vpngate.net/	Python-urllib/1.17
3316252	3/23/13 8:00 PM	198-136-27-242.static.gorillaservers.com	http://www.vpngate.net/en/	Python-urllib/1.17
3316383	3/23/13 8:00 PM	ec2-23-20-4-19.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3316570	3/23/13 8:01 PM	ec2-50-16-163-135.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3317306	3/23/13 8:05 PM	ec2-23-20-4-19.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3317533	3/23/13 8:07 PM	ec2-50-16-163-135.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3318339	3/23/13 8:10 PM	ec2-23-20-4-19.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3318553	3/23/13 8:12 PM	ec2-50-16-163-135.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3319069	3/23/13 8:15 PM	198-136-27-242.static.gorillaservers.com	http://www.vpngate.net/	Python-urllib/1.17
3319072	3/23/13 8:15 PM	198-136-27-242.static.gorillaservers.com	http://www.vpngate.net/en/	Python-urllib/1.17
3319236	3/23/13 8:15 PM	ec2-23-20-4-19.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3319480	3/23/13 8:17 PM	ec2-50-16-163-135.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3320192	3/23/13 8:20 PM	ec2-23-20-4-19.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3320439	3/23/13 8:22 PM	ec2-50-16-163-135.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17
3321185	3/23/13 8:26 PM	ec2-23-20-4-19.compute-1.amazonaws.com	http://www.vpngate.net/en/	Python-urllib/1.17

Figure 13. Spying from Amazon EC2 and Gorilla Servers by the GFW authority.

list into the GFW blocking list. The authority also discharged all blocking of VPN Gate servers. For four days after that, no VPN servers were blocked by the GFW.

March 20: GFW started verifying IP addresses. The authority started verifying IP addresses before inserting them into the GFW blocking list.

April 24: We started collaborative spy detection. We started collaborative spy detection as described in Section 4.3.

6.4. Scalability

As the number of volunteers increases, the total available bandwidth increases. Therefore, the scalability of VPN Gate is bounded by the VPN Gate List Server.

The VPN Gate List Server instance currently consists of three web servers, a database server, a status monitoring server, and a log analysis server. These servers are connected to the Internet via the campus network at our university. Only the web servers receive requests from VPN servers and users. Since these servers execute the same web application, we can easily scale-out their performance.

We use Microsoft SQL Server for our database server, which runs on a PC with an Intel Xeon E3-1230 3.2-GHz processor (Fujitsu PRIMERGY TX100 S3). This PC has 1.0 MB of L2 cache, 32 GB of main memory, and two SSD drives. The load of the database server is lower than the loads of the web servers. The CPU usage of the database server is approximately 5%, while the disk I/O bandwidth is approximately 1.3 MB/s. We estimate that the database server could handle up to 10 times the current load without upgrading the hardware. When this database server becomes overloaded, we can divide VPN servers and clients into several groups and allocate a database server for each group. We think that each group could perform collaborative spy detection independently. This division would also increase availability.

We have found that we can perform status monitoring on 3,000 VPN servers within 10 minutes. We can easily divide this task and allocate subtasks to multiple servers.

6.5. Comparison to Tor

On August 29, 2013, the number of VPN servers was 3,000, which was comparable to the number of Tor nodes. Tor had 4,000 listed relay nodes and 2,000 hidden bridge nodes. We hope that the number of VPN servers in VPN Gate will exceed the number of Tor nodes because we added 500 new servers each in both July and August, 2013.

The number of Chinese users of VPN Gate was larger than that of Tor. At the end of August 2013, we had 9,000 daily unique IP addresses from China, while Tor had an estimated 3,000 daily users from China according to the Tor metrics site[3]. We achieved this result because we focused on bypassing firewalls and implementing collaborative spy detection. In contrast, it is hard for Tor to achieve such collaboration among nodes.

VPN Gate has another advantage over Tor. Since VPN Gate provides a VPN tunnel for IP, a user can use any TCP or UDP application through VPN Gate without having to modify the application or set proxies.

6.6. Problems and discussion

Criminals might use VPN Gate to hide their IP addresses. We can repress such abuse through logging as described in Section 5.4. Another problem is that a volunteer running a VPN server can tap or modify the decapsulated packets of VPN users. This problem is not new. Existing open proxies and Tor exit relays have the same problem, and we currently have no solution to offer. Lastly, a VPN server can potentially use up the bandwidth of a volunteer's Internet line. In response, volunteers can use traffic shaping tools such as NetLimiter[4] to limit the bandwidth of VPN servers.

The innocent IP mixing technique could disturb the owners of innocent IP addresses with users in a country subject to censorship. We have not yet received any complaints from such IP address owners.

Instead of probing VPN servers, a censorship authority could build a whitelist. Maintaining such a whitelist, however, would be quite difficult. Even in a country subject to censorship, Internet access is vital for both residents and visiting businesspeople. One day we might suddenly mix in the IP address of Yahoo! US Mail. The

[3] https://metrics.torproject.org/graphs.html

[4] http://www.netlimiter.com/

Table 5. Monthly transition of the number of GFW authority IP address blocks used for probing.

Year: 2013	IP address blocks (/24)	New blocks	Reused blocks	% reuse
March	2,792	2,792	0	0%
April	2,645	441	2,204	83%
May	1,199	103	1,096	91%
June	1,509	93	1,416	94%
July	1,856	235	1,621	87%
August	1,792	98	1,694	95%
September	1,516	92	1,424	94%
October	1,168	129	1,039	89%

next day we might mix in some important servers for Amazon EC2. Moreover, some servers share the same Akamai or other CDN IP addresses. It would be impossible for a censorship authority to find all important hosts and put their IP addresses into a whitelist in advance.

A censorship authority could also run fake VPN Gate servers to paralyze the VPN Gate network. Such fake servers could send fake logs with false IP addresses to the VPN Gate List Server in order to induce errors in our collaborative spy detection. The false IP addresses could include the valid IP addresses of innocent users. A small number of fake servers cannot impact the entire network, because we can ignore such a small number of false IP addresses. If a censorship authority ran many fake servers, however, these could impact the network. It would be very costly to run so many servers. We assume that a censorship authority would not be willing to pay for such an active attack.

Well-budgeted censorship authorities, like the GFW authority, probably have a large number of IP address blocks available for probing sources. Such IP connectivity infrastructure, however, should have long-term assigned static IP address blocks. Such IP address blocks cannot change frequently. Table 5 lists the actual detected numbers of probing source IP addresses operated by the GFW authority in part of 2013. According to this data, every month the GFW authority reused most of the IP address blocks that had appeared in the previous month. This implies that the GFW authority has only about 4,000 IP address blocks as fixed infrastructure.

6.7. Updated experiences

Since the submission of this paper to NSDI, the following events have happened.

On September 2, 2013 the blocking function of the GFW against VPN Gate became unstable. First, the GFW suddenly stopped blocking all the VPN Gate servers. A few hours later, the GFW recovered and began blocking VPN Gate again. This alternation between blocking and non-blocking continued for a few days.

Since September 5, the GFW has completely stopped blocking all the VPN Gate servers while continuing the probing activity. All servers were reachable through the GFW from September 5, 2013 to February 4, 2014. We do not know why the GFW stopped the blocking.

The number of users and volunteers has increased continuously. On February 4, 2014 we had 5,200 daily volunteers, 1,049,000 daily connections (156,000 unique IP addresses) worldwide, and 123,000 daily connections (16,000 unique IP addresses) from China.

7. Conclusions

We have designed and implemented VPN Gate, a VPN relay system with strong blocking resistance to censorship firewalls such as China's Great Firewall (GFW). In VPN Gate, we use two key techniques to achieve blocking resistance: innocent IP mixing, and collaborative spy detection. We have achieved a proportion of 60-70% of VPN servers not blocked by the GFW. Users in a country subject to censorship can bypass a firewall if they can reach at least one unblocked VPN server. Censorship authorities must block all VPN servers, and this is a very hard task.

VPN Gate works effectively because it relies on many volunteers. We have spent nothing on providing VPN relaying functions. Instead, distributed volunteers contribute small amounts of their electric power and line bandwidth. In contrast, censorship authorities must build expensive censorship infrastructures, implement complex probing programs, and operate them at all times.

The tension between stronger blocking and stronger blocking resistance is essentially a cat-and-mouse game. It is not a fair game, however, and blocking resistance has advantages over blocking. After launching VPN Gate, we played this game with the GFW authority, and we have won the game for the moment. In the future, we are ready to improve our blocking resistance.

In the future, we would like to improve the scalability of the VPN Gate List Server. Additionally, we plan to support IPv6 in VPN Gate.

Note that we did not violate any laws in Japan, where we performed all studies, research, and implementation of blocking resistance to foreign censorship firewalls.

8. Acknowledgements

We would like to thank the NSDI reviewers, our shepherd Professor Sharon Goldberg, Professor Kozo Itano, Professor Kazuhiko Kato, Professor Hisashi Nakai, Professor Akira Sato, Professor Takahiro Shinagawa, Doctor Hiromitsu Takagi, Doctor Tetsuo Sugiyama, Doctor Junpei Kuwana, Doctor Mitsuo Yoshida, Takao Ito, Mei Sharie Ann Yamaguchi, Satoshi Matsumoto, Genya Hatakeyama, Christopher Smith, the Academic Computing and Communications Center of University of Tsukuba, and the National Police Agency of Japan. We are also grateful to the many volunteers and users of VPN Gate.

9. References

[1] Daniel Anderson: "Splinternet Behind the Great Firewall of China", ACM Queue, Vol. 10, No. 11, 2012.

[2] "Azureus User Guide", http://azureus.sourceforge.net/.

[3] Oliver Berthold, Hannes Federrath, and Stefan Koopsell: "Web MIXes: A system for anonymous and unobservable Internet access", Designing Privacy Enhancing Technologies, Springer LNCS 2009, pp 115-129, 2001.

[4] "BitTorrent User Manual", http://www.bittorrent.com/help/manual/.

[5] Richard Clayton, Steven J. Murdoch, and Robert N. M. Watson: "Ignoring the Great Firewall of China", In the Proceedings of the Sixth Workshop on Privacy Enhancing Technologies (PET 2006), pp.20-35, 2006.

[6] Bram Cohen: "Incentives build robustness in BitTorrent", In Proceedings of the Workshop on Economics of Peer-to-Peer Systems, pp.68-72, 2003.

[7] Roger Dingledine, Nick Mathewson, and Paul Syverson: "Tor: the second-generation onion router", In Proceedings of the 13th conference on USENIX Security Symposium, 2004.

[8] Nick Feamster, Magdalena Balazinska, Greg Harfst, Hari Balakrishnan, and David Karger: "Infranet: Circumventing Web Censorship and Surveillance", In the Proceedings of the 11th USENIX Security Symposium, August 2002.

[9] Nick Feamster, Magdalena Balazinska, Winston Wang, Hari Balakrishnan, and David Karger: "Thwarting web censorship with untrusted messenger discovery", In Proceedings of the 3rd Workshop on Privacy Enhancing Technologies (PET 2003), Springer LNCS 2760, pp. 125-140, 2003.

[10] "Free VPN Info and PC Tips @ VpnSurfing.com", vpnsurfing.com. [Online]. Available: http://www.vpnsurfing.com/. [Accessed: 04-Sep-2013].

[11] David Isaac Wolinsky, Kyungyong Lee, P. Oscar Boykin, Renato Figueiredo: "On the design of autonomic, decentralized VPNs", 6th International Conference on the Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010.

[12] Stefan Köpsell and Ulf Hilling: "How to achieve blocking resistance for existing systems enabling anonymous web surfing", In Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2004), 2004.

[13] David Martin and Andrew Schulman: "Deanonymizing users of the SafeWeb anonymizing service", Proceedings of the 11th USENIX Security Symposium, 2002.

[14] Damon McCoy and Jose Andre Morales and Kirill Levchenko: "Proximax: A measurement based system for proxies dissemination," Financial Cryptography and Data Security, 2011.

[15] "PrivacyProtectorGVN", Privacy Protector. [Online]. Available: https://privacyprotector.eu/technology/. [Accessed: 04-Sep-2013].

[16] “rbox-tor: an easy to use Tor server”, redct. [Online]. Available: http://redct.info/rbox/tor.html. [Accessed: 13-Sep-2013].

[17] "Tor Project: obfsproxy", torproject.org. [Online]. Available: https://www.torproject.org/projects/obfsproxy. [Accessed: 04-Sep-2013].

[18] Tim Wilde: "Great Firewall Tor probing Circa 09 Dec 2011." [Online]. Available: https://gist.github.com/da3c7a9af01d74cd7de7. [Accessed: 04-Sep-2014].

[19] Philipp Winter and Stefan Lindskog: "How the great firewall of china is blocking Tor," Proceedings of the 2nd USENIX Workshop on Free and Open Communications on the Internet, 2012.

[20] Eric Wustrow, Scott Wolchok, Ian Goldberg, and J. Alex Halderman: "Telex: Anticensorship in the network infrastructure", In the Proceedings of the 20th USENIX Security Symposium, August 2011.

[21] Xueyang Xu, Z. Morley Mao and J. Alex Halderman: "Internet censorship in China: Where does the filtering occur?", In Proceedings of the 12th International Conference on Passive and Active Measurement (PAM 11), pp. 133-142, 2011.

[22] Chenfeng Vincent Zhou, Christopher Leckie, and Shanika Karunasekera: "A survey of coordinated attacks and collaborative intrusion detection", Computers & Security, Vol.29, No.1, pp.124-140. 2010.