Censorship Resistance as a Side-Effect

Henry Tan
Georgetown University
Washington, DC USA

Micah Sherr
Georgetown University
Washington, DC USA

Abstract

This position paper presents the following thought experiment: can we build communication protocols that (1) are sufficiently useful that they achieve widespread adoption as general-purpose communication mechanisms and (2) thwart censorship as a consequence of their design? We posit that a useful communication platform that is inherently resistant to traffic analysis, if widely adopted and *used primarily for purposes not related to censorship circumvention*, may be too politically and economically costly for a government to block.

1 Introduction

The privacy enhancing technologies community has proposed a number of systems for circumventing government censorship, some of which (notably, Tor [3]) are in active use today. Many existing approaches construct *covert communication channels* that are hidden from the censor's view. For instance, Infranet [4] constructs a covert channel using sequences of seemingly "benign" HTTP requests, Collage [2] embeds messages in images uploaded to sites that host user-generated content, and *decoy routing* techniques such as Telex [17] hide requested URLs in SSL/TLS handshakes. More recently, a number of *traffic shaping* approaches have been proposed (e.g., SkypeMorph [8] and Freewave [6]) that attempt to conceal covert channels by either tunneling them within permitted protocols or changing their traffic patterns to cause them to appear as benign streams.

While the above techniques certainly make censorship more difficult, their security properties are not currently well-understood. In particular, a knowledgeable and powerful censor could potentially defeat such measures by applying steganographic detection techniques [9], enumerating the location of decoy routers [10], and/or leveraging machine learning-based traffic analyzers to perform traffic classification (cf. [14–16]).

Fully understanding the security of existing censorship resistant techniques is an open question that we do not address in this paper. In this position paper, we posit that the security analyses of censorship circumvention systems will likely follow the typical security "arms race" in which discovered vulnerabilities are followed by proposed fixes. Arguably, given the asymmetry between the adversary (e.g., a nation state with centralized control over the nation's communication architecture) and the user of the anti-censorship system (e.g., a dissident who is dependent on the monitored network infrastructure), the advantage in this arms race likely lies with the censor.

This paper takes the position that rather than existing as a standalone system, censorship-resistance should be a characteristic of a widely fielded and general-purpose communication platform. That is, we assert that it is more difficult for a censor to block a ubiquitous and widely-used communication protocol than a niche application designed solely to circumvent censorship. Our goal is to avoid the censor vs. anti-censorship arms race by instrumenting a reliable and high-performance communication primitive that we hope will be widely deployed, *not* used primarily as an anti-censorship apparatus, but that is inherently difficult to surveil and block as a natural consequence of its design.

Paradoxically, to be effective as an anti-censorship technology, such an architecture should achieve widespread adoption for purposes *unrelated* to censorship circumvention. If the primary purpose of the architecture is censorship circumvention, the cost to the adversary of barring access to protocols built using the architecture is low. However, if the architecture is also regularly used for business and commerce, blocking an otherwise useful tool that has widespread adoption may be too politically and economically costly for a censor. To this end, the architecture must both encourage general purpose usage and be competitive with existing methods of communication.

2 Censorship-Resistant Communication Architectures

We consider two parties, Alice and Bob, who want to communicate with each other over the Internet. Eve, the censor, observes and controls all packets going to or coming from Alice. Alice is motivated to prevent Eve from discovering that she is attempting to communicate with Bob. We assume that Bob is outside of the censor's view.

To facilitate its general use as a communication platform and not just as a censorship countermeasure, our architecture should provide benefits over direct IP communication. Below, we briefly outline general-purpose centralized (Section 2.1) and decentralized (Section 2.2) architectures that enable efficient and reliable communication and are also resistant to censorship.

2.1 Centralized Architecture

We observe that, in principle, anti-censorship can be straightforwardly achieved by using a trusted third party to bridge a connection between Alice and Bob, so long as the censor does not block access to the third party. The third party server, which we call the *broker*, maintains full control of the communication network and manages key distribution and status information. We assume that users know the public key of the broker and can hence communicate privately with it. Users upload their public keys to the broker and are required to register with the broker before they can participate in the network. The broker serves as a relay for all communication between clients.

To achieve end-to-end communications privacy, Alice and Bob can query the broker for the other party's public key (certificate) and communicate privately over SSL/TLS, using the broker as an intermediary (i.e., a router). Importantly, messages should be protected using SSL/TLS with the broker so that the censor cannot discover with whom Alice is communicating.

We emphasize that *such a rendezvous mechanism also enhances reliability* since it enables two parties to communicate even when direct IP communication is not available (e.g., when the receiver is behind a firewall or NAT and cannot accept incoming connections). A broker with sufficient resources to provide high bandwidth, low-latency communication between nodes could encourage widespread utilization of the service. Importantly, since Bob's identity is encrypted and (by assumption) Bob is located outside of the censor's view, then the censor cannot distinguish between streams that should be subject to censorship and those that should not. That is, it is left with the choice of either blocking access to the broker — and hence "censoring" *everything* — or permitting all traffic. If sufficiently widely adopted for business and commerce, we posit that the financial cost of blocking the service may outweigh the adversary's desire to censor.

We note that such a centralized architecture is feasible even at large scale, as is illustrated by Google's Voice and Hangout services. However, a centralized design comes with the obvious weakness of having a single global point of failure: should the centralized service be compromised by the censor, attacks such as monitoring, eavesdropping, and censorship become much easier to perform. As indicated by the Snowden documents, governments can (and do) leverage the centralization of existing communication systems (e.g., Skype, Facebook, Google, etc.) to focus their surveillance efforts, with or without the cooperation of the operators of the centralized systems [5].

2.2 Distributed Architecture

We briefly sketch a distributed communication protocol that is performant, has several potentially useful advantages over direct IP communication, and is naturally resistant to monitoring and censorship. Since a major goal of *censorship-resistance by side-effect* is to gain widespread adoption of our protocol, we aim to support a variety of network applications (e.g., voice-over-IP, file transfer, interactive messaging, etc.).

Our protocol makes use of a fully decentralized directory service that supports $\mathrm{put}(\mathrm{key}, \mathrm{value})$ and $\mathrm{value} \leftarrow \mathrm{get}(\mathrm{key})$ semantics. A standard DHT (e.g., Chord [12]) that supports low-cost lookups is a reasonable implementation. When *nodes* (potential communicants) come online, they register by putting their public key as well as a *contact point* into the decentralized directory, keyed by a unique identifier (UID) such as a hash[1] over their email address. To anchor trust in the system, public keys could be signed by peers, creating a social web of trust similar to that used by PGP/GnuPG. Additionally, decentralized certificate verification techniques (e.g., Google's Certificate Transparency [7]) that rely on append-only data structures may provide useful protections.

If a node can receive network communication—e.g., it is not behind a firewall, proxy, or NAT—then it advertises its network address as its contact point. Otherwise, the node (i) chooses a peer as a rendezvous point (RP) and sets its contact point to be the RP's UID, and (ii) creates a TLS connection to its RP.

[1] The use of the hash function provides some privacy protections, since it makes it more difficult to cull email addresses and network locations from the directory.

When a node, Alice, wants to send a message to a node Bob, it queries the directory to discover Bob's contact point and public key. (We assume Alice has apriori knowledge of Bob's UID/email address.) If the contact point is a network address, then Alice initiates direct communication; otherwise, Alice must iteratively query the directory until she learns of an appropriate rendezvous point for Bob. Using the public keys retrieved from the directory, Alice initiates a TLS connection to Bob or Bob's rendezvous point (or the rendezvous point's RP, etc.). In the latter case, Bob's RP relays the communication (again, using a TLS connection) to Bob.

Our envisioned protocol supports *explicit redirection*—the metadata of a message may contain instructions to forward that message to another party. Since messages are encrypted in TLS, this permits a form of onion routing [13] similar to that used by Tor [3].

The above RP and redirection schemes provide useful reachability properties: Alice can contact Bob, regardless of their network locations. That is, Alice can initiate a connection to Bob, even if Bob is behind a firewall or NAT, eliminating the need to develop specialized NAT piercing techniques. In addition to enabling anonymous communication, explicit redirection also improves reachability and reliability, since traffic can be easily rerouted around network failures. And importantly, by adopting the above protocol, developers do not need to build their own directory services, significantly decreasing development time.

To provide high-performance messaging, our protocol can natively take advantage of previously proposed network performance optimization techniques. For examples, the protocol could apply pre-fetching techniques such as SPDY [11] to request multiple objects (e.g., elements of a webpage) in an initial request, reducing the number of roundtrips and significantly shortening latency. Our protocol could also borrow techniques from resilient overlay networks [1] and exploit triangle inequalities in the network underlay to decrease e2e latency and potentially improve goodput.

We argue that the above design—while admittedly far from complete—provides useful properties to application designers, and has the potential to significantly decrease development time. Although the protocol is *not* robust against blocking (in particular, an adversary can prevent access to the directory service), its use of encrypted payloads and potential redirection makes it difficult for an adversary to discern the endpoints and content of an intercepted communication. The censor thus has to choose between preventing all use of the protocol or allowing the protocol's use. If the protocol is sufficiently advantageous to developers and is widely adopted by a variety of network applications, then the adversary may be forced to forgo censorship.

3 Conclusion

This paper proposes two general-purpose communication protocols that inherently resist censorship. To motivate adoption *even when censorship resistance is not a goal*, our protocols are generally useful: they allow peers to communicate when direct IP connections are unsupported (e.g., due to a firewall, proxy, or NAT), and they provide message confidentiality through end-to-end encryption. This paper argues that if such communication designs are widely used, then censors must choose between significant "overblocking" (thus incurring high political and potentially economic costs) and allowing unfettered access to information.

Acknowledgments

This work is partially supported by NSF CAREER CNS-1149832 and NSF grants CNS-1064986, CNS-1204347, and CNS-1223825. The findings and opinions described in this paper are those of the authors, and do not necessarily reflect the views of the National Science Foundation. Additionally, this material is based upon work supported by the Defense Advanced Research Project Agency (DARPA) and Space and Naval Warfare Systems Center Pacific under Contract No. N66001-11-C-4020. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Project Agency and Space and Naval Warfare Systems Center Pacific.

References

[1] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. The Case for Resilient Overlay Networks. In *Workshop on Hot Topics in Operating Systems (HotOS)*, 2001.

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

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

[4] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. Karger. Infranet: Circumventing Web Censor-

ship and Surveillance. In *USENIX Security Symposium (USENIX)*, 2002.

[5] B. Gellman and A. Soltani. NSA Infiltrates Links to Yahoo, Google Data Centers Worldwide, Snowden Documents Say. The Washington Post, October 30 2013.

[6] A. Houmansadr, T. Riedl, N. Borisov, and A. Singer. I Want My Voice to be Heard: IP over Voice-over-IP for Unobservable Censorship Circumvention. In *Network and Distributed System Security Symposium (NDSS)*, 2013.

[7] B. Laurie, A. Langley, and E. Kasper. Certificate Transparency. RFC 6962, Internet Engineering Task Force, 2013.

[8] H. M. Moghaddam, B. Li, M. Derakhshani, and I. Goldberg. SkypeMorph: Protocol Obfuscation for Tor Bridges. In *ACM Conference on Computer and Communications Security (CCS)*, 2012.

[9] N. Provos and P. Honeyman. Detecting Steganographic Content on the Internet. Technical Report 01-11, Center for Information Technology Integration, University of Michigan, 2001.

[10] M. Schuchard, J. Geddes, C. Thompson, and N. Hopper. Routing Around Decoys. In *ACM Conference on Computer and Communications Security (CCS)*, 2012.

[11] SPDY: An Experimental Protocol for a Faster Web. `http://www.chromium.org/spdy/spdy-whitepaper`.

[12] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, 2001.

[13] P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous Connections and Onion Routing. In *IEEE Symposium on Security and Privacy (Oakland)*, 1997.

[14] A. M. White, K. Snow, A. Matthews, and F. Monrose. Phonotactic Reconstruction of Encrypted VoIP Conversations: Hookt on fon-iks. In *IEEE Symposium on Security and Privacy (Oakland)*, 2011.

[15] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson. Spot Me if You Can: Uncovering Spoken Phrases in Encrypted VoIP Conversations. In *IEEE Symposium on Security and Privacy (Oakland)*, 2008.

[16] C. V. Wright, L. Ballard, F. Monrose, and G. M. Masson. Language Identification of Encrypted VoIP Traffic: Alejandra y Roberto or Alice and Bob? In *USENIX Security Symposium (USENIX)*, 2007.

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