Video In-Coding: Hiding Arbitrary Data in Video Chat

Paper # TODO: *xxx*

ABSTRACT

Censoring regimes often have a set of network protocols that they permit. In particular, even the most restrictive regions tend to offer some form of video or audio chat so that their citizens may keep in touch with friends and family abroad. Several systems have been proposed to send arbitrary (possibly censored) data over chat programs' audio channels. This is a promising approach, but it restricts overall throughput to those achievable by software modems—on the order of tens of kilobits per second. To date, it has remained an open question as to whether video could also be used for surreptitious communication.

We present Video In-Coding (VIC), a set of techniques for encoding and decoding data over popular video chat systems such as Skype and Google Hangouts. Our insight... Like recent approaches to protocol obfuscation, VIC makes no modifications to the underlying video chat system itself, treating it instead as a black box. Unlike recent approaches, VIC achieves significantly higher throughput, making it a viable and usable tool towards fighting censorship.

1. INTRODUCTION

A powerful censor is able to block traffic to or from a set of hosts [12, 1], block packets with particular "keywords" [3], and even block entire protocols, such as Tor [27, 23, 21]. However, most censoring regimes allow *some* applications to freely communicate with those outside their borders. In such a regime, a promising approach is to "obfuscate" the messages and behavior of a forbidden protocol so that it appears to a passive or active observer as if it were an allowed protocol.

The foremost challenge in protocol obfuscation is that the forbidden protocol must look and *act* like the allowed protocol—not only in terms of packet sizes [29] and timing, but also with respect to how the common implementations of the allowed protocol react to, say, delayed or malformed packets [4, 7]. Thus, although most prior work in protocol obfuscation has focused on trying to mimic the allowed protocol [25, 17, 26], others have demonstrated that subtle differences in their implementations reveal them as being fakers [4, 7].

Rather than pursue a cat-and-mouse game between protocol obfuscation and discovery inherent to mimicry, we propose to investigate the less-studied approach of *using the allowed protocol itself* as a covert channel to carry traffic that would otherwise have been forbidden. Prior work in this space has faced a trade-off between throughput and the flexibility in the types of applications they can support. FreeWave [18] makes arbitrary traffic appear to be Skype traffic by encoding it with a software modem and sending the audio through Skype as if it were a voice call—FreeWave is limited in throughput, demonstrating a maximum rate of 19.2 kbps. Conversely, Facet [14] achieves greater capacity by using Skype's video chat, but is constrained to video streaming applications.

We propose to develop protocol obfuscation techniques to simultaneously achieve high throughput and support arbitrary applications. In line with previous work, we focus our study on Skype, which many censors tend to permit. Specifically, we propose to develop techniques to (1) achieve a high-throughput, covert channel on top of Skype video, and (2) establish resilient connections from a host within a censored regime to hosts outside. If successful, these techniques will effectively permit "a line out" of an attacker's regime, and can thus serve as a building block towards the resilient end-to-end applications that we explore in our other research tasks.

2. BACKGROUND

3. SKOR DESIGN

Our insight is that we can use Skype's video chat—a high-throughput channel—to send arbitrary data by applying recent results in representing arbitrary data with pictures, such as a QR code. Figure 1 shows a high-level design of an obfuscating transceiver that takes arbitrary input data (e.g., Tor or Web traffic), and constructs a pictorial representation of the data (e.g., a barcode or QR Code) that it provides to Skype as if it were video from a web cam. Arbitrary applications communicate with it via a virtual network interface, that is, it requires no modifications to nor knowledge of any specific applications. Facet [14] hides within Skype's video channel, but works only for exchanging video (e.g., from YouTube), and not for arbitrary data. The fundamental difference with our goal is that Facet's transmission is lossy—for video, this may suffice, but supporting *arbitrary* applications requires the ability to detect and recover from inevitable losses.

There has been a recent surge in the study of Visible Light Communication (VLC), wherein (typically mobile devices') display screens act as "transmitters" and cameras act as "receivers," permitting simple point-to-point communication between two nearby users [15, 19, 20, 24, 9]. Because the goal setting of this space involves two users holding their phones at one another, much of this work focuses on developing efficient pictorial representations of data that are

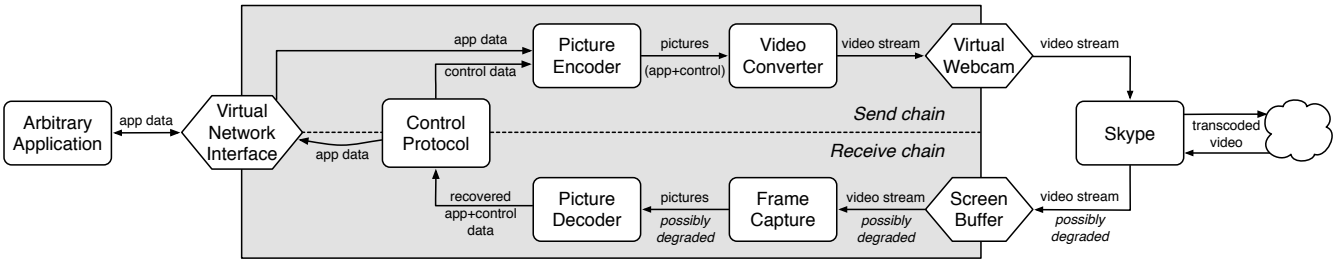

Figure 1: A high-level design of protocol obfuscating through Skype video. From the perspective of the transceiver, Skype is a "black box" that can introduce noise by transcoding our video. The Picture Encoder reacts to this by altering the pictures' resolution, effective frame rates, and so on, and is guided by the Control Protocol.

resilient to skew, translation, rotation, and lack of focus [19, 15, 20, 9]. Our proposed setting differs fundamentally from prior work in that we can assume the picture is always centered properly and not rotated, and thus we believe that we can develop unique encoding techniques to achieve greater throughput.

We are similar to prior settings in that we, too, must correct for a noisy channel, though our sources of noise are quite different. One source of noise is the standard loss of packets between the two Skype clients, but there is another, more pervasive noise source: Skype itself. Skype transcodes its video to try to match the available capacity in order to balance high-quality and smoothness. From our transceiver's perspective, this equates to blurriness, possibly dropped frames, and even partial repetition of an old frame within a new one. As Skype's transcoding is dynamic, so too must our coding be in order to achieve as high throughput as possible.

We therefore include in our design a dynamic control algorithm that will monitor video quality at both ends, and exchange this information so as to dynamically adjust the rate at which we send data. As a first approximation, this is similar in spirit to TCP's rate control, but strictly speaking, we cannot control the *rate* at which packets leave the sending client—ultimately, Skype is sending. Instead, what we can control are the various parameters of the pictures and video we feed as input to Skype: the resolution (e.g., the size of the boxes in a QR code), and the effective frame rate (how many frames we dedicate to a given picture in the video stream we feed to Skype). One of the main sources of noise will be blurriness; to detect how high a resolution a receiving client can handle, we will include in our pictorial messages a sort of "eye chart," like those available in an optometrist's office—the receiving client then replies with "how far down" the chart it can clearly decode.

It may be possible that some portions of a frame are decodable, while others are too blurry. We will apply techniques from wireless networking, known as *partial packet recovery* [6, 10, 16], which allow end-hosts to retransmit strict subsets of packets, rather than retransmit the parts of a packet that were successfully decoded at the receiver. We believe that translating these findings to a two-dimensional, pictorial setting will be broadly applicable to other VLC work.

Through optimizations such as these, our proposed transceiver will, if successful, achieve far greater throughput than prior obfuscating systems that permit arbitrary communication. To the best of our knowledge, the only successful systems that send arbitrary data through Skype use its audio channel, and thus are limited to modem speeds [29, 7, 4]; for instance, FreeWave [18] reports a maximum throughput of 19.2 kbps. A critical finding from this prior line of work is that, although Skype encrypts its traffic, it does not use fixed-sized packets, and thus it is possible for an eavesdropper to make surprising inferences based only on the distribution of packet sizes. In particular, by simply observing packet sizes, one can infer what language two Skype users are speaking [29], and infer phrases [28]. Our design must take this into account, as well; though we have limited control over packet sizes, we can influence them by, e.g., ensuring that there is not so much entropy between frames that Skype's codec believes we need a higher fraction of key frames than typical Skype video. One possible approach to mitigating this is to blend our data with more traditional video (e.g., faces); alternatively, we could dedicate a fraction of our video to standard video, and place our encoded information along the border. Balancing performance and entropy will be a critical component of this research task, but we believe that even conservative levels of performance will far exceed those achievable through audio alone.

4. WHOM AND HOW TO CALL

The above design outlines our proposal for achieving high-throughput communication over a video chat between two Skype clients, but there are several key questions that are critical to the security of the system. First, even if we are successful at obfuscating arbitrary protocols as Skype video, the attacker can observe call durations and frequency. As a simple example, consider a user browsing the web from within a censoring regime; if we were to create a new Skype video call on a page-by-page basis, then to the attacker this would appear to be many short-lived video calls—because users are unlikely to use Skype in such a fashion, this could leak that the user is communicating covertly. We will investigate how often users should set up and tear down connections to external peers. One promising approach is to involve the user in this process by making explicit that the system should only be used, say, an hour a day, and for mostly contiguous chunks of time. The system would have to support sending chaff traffic during times that the user is not sending or receiving data; fortunately, this is already built into our design, as the Picture Encoder generates chaff pictures at fixed intervals, whether it has data to send or not.

Second, we will investigate whom a user should call to establish this covert channel. Prior work has proposed includ-

ing a list of Skype IDs in the code deployment, but it is not immediately clear if this suffices in the presence of a powerful adversary. Skype is generally known to encrypt its traffic and to use Super Nodes as intermediaries between communicating users (if those users are both behind NATs) [2, 5]. If a user within a censored regime connects to a host outside through a Super Node that is also outside the censored regime, then the censor would not be able to distinguish this from standard Skype traffic, and would let it through. However, suppose the censor were capable of running a Super Node itself: it would then be able to observe the Skype IDs on both sides of the communication, compare this to the publicly known list of covert Skype IDs, and simply drop the traffic.

To address this, we propose to develop techniques for a secure "control plane" for such communication. One possible approach is to allow users to request that the intermediary nodes are outside of a particular regime, not unlike the PIs' prior work on route avoidance [13]. Another possible (and complementary) approach is to not publicize the set of covert Skype IDs, and to instead randomize them in a fashion that end-users can compute locally. A critical feature, and technical challenge, to this approach is to develop a scheme of randomizing Skype IDs that is easy for a user to compute and try, but difficult for an observer to quickly determine whether or not a given Skype ID maps to a covert channel. This is similar in some sense to decoy routing [11, 30, 8, 22], but is distinguished by the fact that Skype IDs come from a far larger space than IPv4 addresses, and can be easily generated and allocated.

5. CONCLUSION

Acknowledgments

6. REFERENCES

[1] D. Anderson, "Splinternet behind the great firewall of China," *Queue*, vol. 10, no. 11, Nov. 2006.

[2] S. A. Basset and H. G. Schulzrinne, "An analysis of the Skype peer-to-peer Internet telephony protocol," in *IEEE Conference on Computer Communications (INFOCOM)*, 2006.

[3] R. Clayton, S. J. Murdoch, and R. N. M. Watson, in *Workshop on Privacy Enhancing Technologies (PET)*, 2006.

[4] J. Geddes, M. Schuchard, and N. Hopper, "Cover your ACKs: Pitfalls of covert channel censorship circumvention," in *ACM Conference on Computer and Communications Security (CCS)*, 2013.

[5] S. Guha, N. Daswani, and R. Jain, "An experimental study of the Skype peer-to-peer VoIP system," in *Workshop on Peer-to-Peer Systems (IPTPS)*, 2006.

[6] B. Han, A. Schulman, F. Gringoli, N. Spring, B. Bhattacharjee, L. Nava, L. Ji, S. Lee, and R. Miller, "Maranello: Practical partial packet recovery for 802.11," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2010.

[7] A. Houmansadr, C. Brubaker, and V. Shmatikov, "The parrot is dead: Observing unobservable network communications," in *IEEE Symposium on Security and Privacy*, 2013.

[8] A. Houmansadr, G. T. K. Nguyen, M. Caesar, and N. Borisov, "Cirripede: Circumvention infrastructure using router redirection with plausible deniability," in *ACM Conference on Computer and Communications Security (CCS)*, 2011.

[9] S. Hranilovic and F. R. Kschischang, "A pixelated MIMO wireless optical communication system," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 12, no. 4, pp. 859–874, 2006.

[10] K. Jamieson and H. Balakrishnan, "PPR: Partial packet recovery for wireless networks," in *ACM SIGCOMM*, 2007.

[11] J. Karlin, D. Ellard, A. W. Jackson, C. E. Jones, G. Lauer, D. P. Mankins, and W. T. Strayer, "Decoy routing: Toward unblockable Internet communication," in *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2011.

[12] S. Khattak, M. Javed, P. D. Anderson, and V. Paxson, "Towards illuminating a censorship monitor's model to facilitate evasion," in *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2013.

[13] D. Levin, Y. Lee, L. Valenta, Z. Li, V. Lai, C. Lumezanu, N. Spring, and B. Bhattacharjee, "Alibi routing," in *ACM SIGCOMM*, 2015.

[14] S. Li, M. Schliep, and N. Hopper, "Facet: Streaming over videoconferencing for censorship circumvention," in *Workshop on Privacy in the Electronic Society (WPES)*, 2014.

[15] R. LiKamWa, D. Ramirez, and J. Holloway, "Styrofoam: A tightly packed coding scheme for camera-based visible light communication," in *ACM Workshop on Visible Light Communication Systems (VLCS)*, 2014.

[16] K. C.-J. Lin, N. Kushman, and D. Katabi, "ZipTx: Harnessing partial packets in 802.11 networks," in *ACM Conference on Mobile Computing and Networking (MobiCom)*, 2008.

[17] 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.

[18] H. M. Moghaddam, T. Reidl, 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.

[19] S. D. Perli, N. Ahmed, and D. Katabi, "PixNet: Interference-free wireless links using LCD-camera pairs," in *ACM Conference on Mobile Computing and Networking (MobiCom)*, 2010.

[20] N. Rajagopal, P. Lazik, and A. Rowe, "Hybrid visual light communication for cameras and low-power embedded devices," in *ACM Workshop on Visible Light Communication Systems (VLCS)*, 2014.

[21] Reporters Without Borders, "Enemies of the Internet 2013 Report," https://surveillance.rsf.org/en/wp-content/uploads/sites/2/2013/03/enemies-of-the-internet_2013.pdf, Mar. 2013.

[22] M. Schuchard, J. Geddes, C. Thompson, and N. Hopper, "Routing around decoys," in *ACM Conference on Computer and Communications Security (CCS)*, 2012.

[23] Tor Metrics, "https://metrics.torproject.org."

[24] Q. Wang, D. Giustiniano, and D. Puccinelli, "OpenVLC: Software-defined visible light embedded networks," in *ACM Workshop on Visible Light Communication Systems (VLCS)*, 2014.

[25] Q. Wang, X. Gong, G. T. Nguyen, A. Houmansadr, and N. Borisov, "CensorSpoofer: Asymmetric communication using IP spoofing for censorship-resistant web browsing," in *ACM Conference on Computer and Communications Security (CCS)*, 2012.

[26] Z. Weinberg, J. Wang, V. Yegneswaran, L. Briesemeister, S. Cheung, F. Wang, and D. Boneh, "StegoTorus: A camouflage proxy for the Tor anonymity system," in *ACM Conference on Computer and Communications Security (CCS)*, 2012.

[27] P. Winter and S. Lindskog, "How the great firewall of China is blocking Tor," in *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2012.

[28] 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*, 2008.

[29] 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*, 2007.

[30] E. Wustrow, S. Wolchok, I. Goldberg, and J. A. Halderman, "Telex: Anticensorship in the network infrastructure," in *IFIP International Information Security and Privacy Conference (SEC)*, 2011.