TRIST: Circumventing Censorship with Transcoding-Resistant Image Steganography

Christopher Connolly, Patrick Lincoln, Ian Mason, Vinod Yegneswaran
connolly@ai.sri.com, {lincoln, iam, vinod}@csl.sri.com
SRI International

Abstract

We explore the viability of extending state-of-the-art image steganography techniques for bypassing censorship. Our quest for a scalable steganographic technique, which is robust against automated transcoders that reformat images in-flight, led to the implementation of a prototype system called TRIST[1] that embeds data by selectively modifying bits in the frequency domain of the image. By choosing heavily quantized frequency components at low JPEG quality values, we can robustly embed information within images, and demonstrate how this information survives a number of transformations, including transcoding to higher JPEG quality levels and other perturbations, such as image resizing (within bounds).

We evaluate our system by building a prototype of a transcoding-resistant steganography library that we integrate with StegoTorus [36]. Our evaluations demonstrate that StegoTorus integrated with TRIST provides reasonable bandwidth capable of supporting basic web surfing along with transcoding resilience. Finally, we describe how our system can be adapted to counter state-of-the-art statistical attacks such as blockiness detectors.

1 Introduction

Censorship attempts by various countries to block anonymity systems, such as Tor, have precipitated the development of diverse proxy systems that aim to evade censorship by imitating popular protocols such as HTTP [7, 36] and Skype [27]. There are multiple systems that have attempted to use image steganographic techniques to bypass censorship. These include proxy systems such as Infranet [8] and offline systems such as Collage [3] and MIAB [20] that rely on social-media sharing sites like `Flickr` [12] and web blogs to distribute steganographic content. However, these steganographic schemes aren't resilient to basic image transformations routinely performed by many of these sites to optimize storage and bandwidth. Furthermore, a simple and effective means to disrupt the use of such systems involves the deployment of commodity off-the-shelf (COTS) *transcoding proxies* [6, 16, 33] that seek to improve performance by dynamically re-encoding images at lower quality levels and rescaling.

To address these limitations, we propose a new steganographic approach that operates on the frequency-domain of images. By choosing heavily quantized frequency components at low JPEG quality values, we can robustly embed information within images, and this information survives a number of transformations, including transcoding to higher quality, Not surprisingly, when starting at a low base quality level, the message survives transcoding to a higher quality and back to the base quality. Heavily quantized frequency components tend to be stabilized because they can only take on a limited number of values. More interestingly, the embedded message survives image rescaling, as long as the extraction occurs after an inversion of the scaling operation. Depending on the cover image and the frequency components used, the message can survive an image reduction of up to 75%, or an image expansion of up to 150%.

Motivated by these results, we design and implement a prototype general purpose library to facilitate the development of transcoding-resistant steganographic systems. We evaluate the prototype library by extending the StegoTorus pluggable transport with a new JPEG steganography scheme. Our evaluation results indicate that the overhead of our transcoding-resistant JPEG steganography scheme is comparable to that other schemes and does not significantly impact the performance of StegoTorus. We also evaluate the resilience of our scheme to statistical attacks, specifically the blockiness detector using calibration and reembedding that has been proven to be effective against many JPEG steganography schemes. We find that such detectors can be evaded by transcoding the image to higher quality levels before transmission and transcoding back to lower quality before destegging.

Contributions. In summary, the contributions of our paper include the following:

[1] derives from trist/tryst meaning a secret meeting or rendezvous;

System	File Type	Domain	Steganographic Technique	Detection Strategies and Metric
JSteg [35]	JPEG	frequency	LSB encoding	χ^2, histogram symmetry
JP Hide&Seek [25]	JPEG	frequency	random LSB encoding	χ^2, histogram symmetry
F5 [37]	JPEG	frequency	matrix encoding, permutative straddling	calibration, histogram shape
OutGuess [30]	JPEG	frequency	redundant bit encoding	calibration, reembedding, blockiness
HUGO [11]	JPEG	frequency	LSB matching w/ STC	SVM
YASS [34]	JPEG	spatial	randomized embedding	Cartesian calibration
UNIWARD [18]	JPEG	both	universal embedding	

Table 1: Summary of notable prior JPEG steganography systems and steganalysis techniques

1) Presentation of transcoding-resistant steganography as a problem for censorship circumvention; **2)** Development of a steganographic embedding scheme for JPEG in the frequency domain; **3)** Evaluation of the proposed scheme for transcoding resistance properties; **4)** Integration with the StegoTorus pluggable transport and evaluation of system performance; and **5)** Evaluation of system resilience to statistical attacks.

2 Related Work

We broadly categorize prior related work as belonging to four categories and discuss them below.

1) Transcoding Techniques. Transcoding techniques [6, 5, 16, 33] seek to improve bandwidth performance at the expense of quality by dynamically converting multimedia objects from one form to another along the network path. While these studies do not consider transcoding from a censor's perspective, we are informed by the transformations they perform as they are illustrative of the types of COTS tools that might be easily deployed by censoring countries.

2) Steganography Techniques. Table 1 provides a summary of the most popular steganography systems, the specific steganographic techniques that they implement, and detection strategies known to work against them. JSteg [35], JP Hide&Seek [25], F5 [37], and OutGuess [30] embed message bits by manipulating the quantized DCT coefficients. JSteg with random straddling as well as JP Hide&Seek are detectable using the generalized χ^2 attack. Fridrich et al. exploit the fact that F5 predictably affects the shape of the histogram of DCT coefficients [14]. To defeat OutGuess, Fridrich et al. define a new metric, called blockiness, that measures discontinuities along the boundaries of the 8x8 JPEG grid [13]. HUGO [11] implements a variant of LSB matching that uses STCs to minimize pixel distortions. However, it has been shown to be vulnerable to SVM-based classifiers [15]. YASS uses Quantization Index Modulation (QIM) that confuses traditional blind steganalysis schemes by intentionally making no attempt to minimize embedding impact on the cover image [34], but is detectable through Cartesian calibration techniques [23]. Finally, UNIWARD introduces a Wavelet-based universal embedding function for which there is currently no statistical detection algorithm, but is vulnerable to transcoding attempts [18].

3) Watermarking Techniques. In general, watermarking methods are designed to mark the medium, usually redundantly, with a relatively small (in bits) identification key. Watermarking for copyright protection is most concerned with preserving the watermark under a variety of possible image transformations. Hence, watermarking tends to be redundant and has a low bandwidth requirement relative to steganography. Most watermarking methods *add* the watermark to the underlying image representation. Because of quantization effects, it is possible that the relatively small perturbations of the image representation employed by watermarking would be corrupted by transcoding. This might not affect watermarks for the purpose of human visual inspection, but this has an impact on the use of watermarking strategies for relatively higher-bandwidth steganographic communication. In contrast to most watermarking methods, the approach we propose exploits the existing processing chain for JPEG and MPEG to *set* selected frequency coefficients, and exploits the stabilization properties of quantization to improve robustness.

Watermarking in the transform domain first requires the image to be transformed into frequency or some other generalized Fourier domain (e.g., DCT [1, 29, 28], wavelet [32] or Legendre [40]) to exploit invariance or robustness properties that are characteristic of that domain. In addition, most transform-based approaches allow one to minimize the perceptual effect of the watermark. A common approach [1, 29, 28] embeds the watermark using a weighted sum of DCT coefficients. The new image representation C' is given by:

$$C' = \alpha C + \beta W$$

where C is a coefficient of the original source image, W is the corresponding coefficient for the watermark, and α and β are weights that sum to 1. Usually, such schemes apply the transform over the entire image, but the use of the DCT is especially attractive since this transform is used by both JPEG and MPEG. Other basis functions are available, including the Haar wavelet basis [32], and the Legendre basis [40]. In [32], the wavelet transform is applied first, followed by a singular value decomposition for each band, under the assumption that a perturbation of the singular values of the Haar transformed image is robust to certain transformations but also tends to be less perceptible to the human visual system. Otherwise, the watermark is *added* to the image representation as above. This method is, however, an expensive computa-

tion compared to JPEG compression or decompression. In [40], Legendre moments are employed to allow the watermark to be robust with respect to affine transforms of the image. This is particularly useful for embedding watermarks that are designed for human visual inspection. All of these approaches *additively perturb* the image representation with the watermark.

More sophisticated embedding algorithms [4] exploit quantization by noting that the ordering of coefficients is preserved under quantization, hence this property can be used to encode individual bits by forcing a particular ordering across selected frequency components. Our approach supports a somewhat higher bit rate while being robust to a variety of transformations.

4) Circumvention and Anti-Censorship Systems. Collage [3] and MIAB [20] leverage media sharing websites e.g., `flickr.com` and blog sites to to hide messages within user-generated photos. The assumption is that these censors would be hard pressed to block all of these websites. Their prototype implementations rely on OutGuess and HUGO for image steganography and could be substituted with a transcoding-resistant system. Infranet [8] and StegoTorus [36] conceal traffic that would otherwise be blocked within seemingly normal HTTP traffic. The transcoding resistant image steganography techniques that we develop are complementary and could be used to extend these and other circumvention proxies such as Flash proxies [9], Telex [39], Decoy Routing [22], and Cirripede [19].

3 Adversary Model and System Goals

Adversary Model. We assume that the system user is located in a censored country and is using the system to communicate with a remote server outside the censored zone. We assume that the user and remote endpoint have a shared secret that they could leverage to parameterize the embedding of the image. This shared secret could have been obtained through an offline rendezvous process [26, 10, 9].

The goal of the adversary is prevent censorship circumvention by accurately identifying and disrupting any communication that involves the use of steganographic images. We assume that the adversary has deep packet inspection (DPI) capability to eavesdrop on all traffic between the censored user and the remote endpoint. The adversary does not care to decrypt the underlying message (as its often a TLS stream in the case of Tor pluggable transports) and does not have a priori knowledge of the images that would be used to embed steganographic content. The adversary may employ various statistical techniques to distinguish steganographic images from normal images. Finally, the adversaries could use image transcoders to transform all uploaded and downloaded images. While there are many possible transformations that could be applied to images (e.g., blurring, noise additions, rotations etc.), we focus on two common strategies implemented by commodity transcoders: modifying the JPEG compression metric (q value) and the spatial geometry.

System Goals. We describe below the specific design goals of our proposed system:

1. Unobservability – It must be infeasible for an adversary to use automated techniques to distinguish JPEGs created by our system from normal JPEGs. Unlike most prior work on steganography, human perceptability, i.e., non-distortion of the source cover to a visually unacceptable level, is a non-goal of our system.

2. Transcoding Resistance – The system must continue to be able to transmit data even in the presence of an adversary who manipulates images in between the sender and receiver.

3. Usable Performance – The system must provide reasonable bandwidth. Realized bandwidth is directly proportional to the underlying channel capacity or steganographic overhead. Ideally, the steganographic expansion factor should be within an order of magnitude.

4 JPEG Overview

The JPEG image format [21] offers a compact way to store images. It is a lossy compression scheme that saves space by heavily quantizing or even removing the highest spatial frequencies in an image. Quantization and compression is applied independently to successive blocks of an image. For the sake of this paper, we assume without loss of generality that images are grayscale and divided into 8x8 blocks (called "Minimum Coded Units" or MCUs). Each MCU can be treated as a 64-element vector of integers that represent pixel intensities. At a high level, JPEG compression treats each 8x8 MCU in sequence by first computing a discrete cosine transformation (DCT) of the pixel values, quantizing the resulting frequency coefficients to reduce storage requirements while preserving "perceptually significant" image features, and then Huffman coding the result (see Figure 1). JPEG compression is controlled by a quality factor. As quality is lowered, the highest frequencies of the image are more heavily quantized and ultimately removed.

To embed messages, we exploit the fact that JPEG compression quantizes and therefore stabilizes certain frequency components. This in turn can provide a kind of error correction, since the quantization mapping is many-to-one. Noise or corruption in the quantized frequency components of the original image will tend to be stabilized on output by the loss induced by JPEG compression. This allows the message to survive a number of different transcoding and filtering operations.

Our message embedding recipe first converts a cover image I using a quality q into a new JPEG image I'. We then select four heavily quantized DCT frequency components f_u, f_v, f_w, f_x that can support at least two bits after quantization. Each byte of the message is then em-

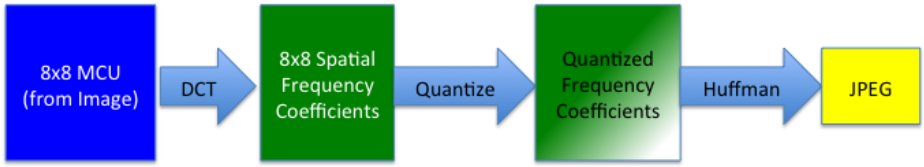

Figure 1: JPEG processing pipeline for compression. The shading in the third box illustrates the effect of quantization on coefficient magnitude, where white = 0. Higher frequencies (in the lower right) are most heavily quantized.

bedded in successive MCUs by splitting that byte into four 2-bit quantities. The corresponding frequency coefficients f_u, f_v, f_w, f_x are set to values that, *after quantization*, will fall in the range [-2,1] for each 2-bit value. This step uses the quantization table that is included with every JPEG image to determine the appropriate target frequency coefficient values. Finally, the result is written emit the result I_M as a JPEG-compressed file at quality q, containing the message M.

Messages are recovered by inverting the recipe, assuming knowledge of the base quality level and the frequency components that were used for embedding (these can be shared secrets). For retrieval, images are first transcoded back to their base quality level, and then each byte is reassembled from each MCU by extracting the values of f_u, f_v, f_w, f_x and assembling the 2-bit quantities into one 8-bit byte. In practice, we use the open-source libjpeg library, version 6b [24]. This library allows direct access to the DCT frequency components for any MCU of a JPEG image, and hence it is straightforward to manipulate the frequency components directly and output the result as a JPEG-compressed file.

Steganographic Expansion Factor. We derive the expansion factor by empirically examining the typical JPEG files after compression at various quality levels. In general, the observed compression at quality 30 results in a 1:6 ratio of message to JPEG file length for the cover image. After embedding, the JPEG file length will often increase, depending on the message content, and can be as much as double the size of the cover JPEG. Thus, we expect anywhere from a 1:6 to 1:12 ratio of message to JPEG length.

5 Robustness Experiments

We performed experiments to help us understand the robustness of this form of message embedding. All of our experiments were performed using the ImageMagick "convert" utility. Messages were constructed by drawing each of 3000 bytes in the message from a uniform distribution over the interval $[0, 255]$. In all experiments, we measured the Hamming distance between the original message and the recovered message. This provides us with an error in bits that characterizes our ability to recover the message through various kinds of transformation. In the first set of experiments, we chose a base quality for embedding, and then transcoded I_M to a new

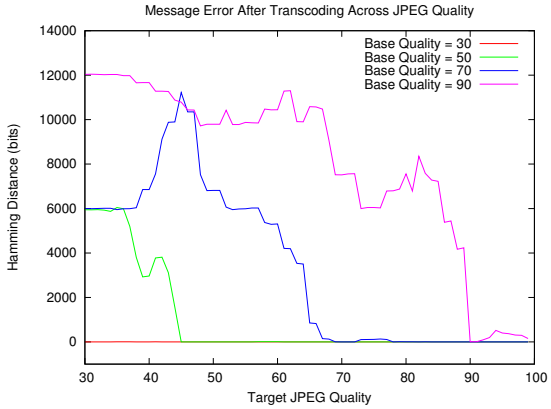

Figure 3: Results of transcoding I_M from a base quality to a target quality and back, using a fixed set of frequency components $(f_{10}, f_9.f_8, f_3)$. By using a base quality of 30 (the red curve), we achieve nearly perfect message transmission over a large range of target quality levels.

target JPEG quality and then back to the base quality, to see whether the message survived changes across quality factors. Figure 3 shows the results, which are generally independent of the image. Note that error rates are very close to zero at and above the base quality. For a base quality of 30, hardly any error is observed on transcoding across quality levels. The strategy suggested by this plot is to embed messages using low frequency components at the lowest quality value that is practical, so that these components are heavily quantized. Transcoding from a low quality to a higher quality and back will not degrade the message.

In a second set of experiments, we applied image rescaling to determine the robustness of message transmission through image enlargement and reduction. Our results were heavily dependent on image characteristics. Highly textured images produced the worst results, most likely because of cross-MCU bleed-through as a result of filtering. By default, image rescaling in ImageMagick relies on two filters that are useful for resampling: the Mitchell filter and the Lanczos filter for image reduction. The support for these filters is 2 or 3 pixels in radius. This means that filtering will cause information to cross MCU boundaries. The effect is greatest at high frequencies, causing significant bleed-through. If we select low frequencies and low quality levels, we can min-

Figure 2: Results of embedding a message in four selected frequency components of an image. Left: "Clean" JPEG images at quality 30; Right: embedding using the highest admissible frequencies;

imize bleed-through across MCUs and at the same time exploit quantization to stabilize the message. The use of Mitchell or Lanczos filters for resampling can, to some extent, be inverted by the use of the "-sharpen" option (essentially a bandpass filter) for "convert". More generally, an impulse response measurement allows us to invert any linear filtering that is present in the transcoding process, so knowledge of the exact form of the filter is unnecessary.

In practice, we can get good message recovery by performing an inverse rescaling operation (to bring the image back to its original resolution), coupled with a sharpening operation. Figure 4 (top) shows one plot using a specific set of frequency components (indices 18, 17, 16, and 10). Error rates are nearly zero when images are scaled by $> 100\%$. At a rescaling of 100%, we see errors simply because the sharpening filter is in use. Recovery would be perfect, or nearly so, if sharpening were omitted in this case. For scale factors $< 100\%$, there is a range of scale factors from about $60-80\%$, but only a narrow range of sharpening sigma within which good error rates are found. Better results are achieved by moving to lower frequencies. In Figure 4 (bottom), the frequency indices are 10, 9, 8, and 3. With these frequencies, message errors are near 0 even in the rescaling range of $75-95\%$, for a wide range of sharpening sigmas.

In general, a good strategy for message embedding is to use the lowest quality that is practical. Our approach to message embedding can tolerate a certain amount of image reduction, but below 70% reduction, error rates increase. In general, redundant coding or some other form of error correction (beyond that provided by JPEG itself) should greatly improve our ability to transmit information through image or video media. In the case of image reduction, we believe that a more thorough study of the properties of resampling filters can help us improve error rates. Finally, we note that the method we have described here is applicable to MPEG, and in particular to I-frame encoding, which is very similar to JPEG processing.

6 System Performance Evaluation

TRIST is implemented as a standalone library in approximately 5900 lines of C code. It extends the widely used libjpeg [24] library for manipulating JPEG images. To evaluate the efficacy and overhead of our JPEG embedding scheme, we integrated TRIST into the StegoTorus pluggable transport as a new steganograpic scheme. The changes necessary to StegoTorus to support this scheme were fairly modest ($\sim$350 lines of C code).

To evaluate the system in a reproducible network environment, we configured StegoTorus as one-hop SOCKS proxy in the localhost and used dummynet [31] to induce a specific one-way link delay ranging from 20-100 ms. We then used curl to connect to a local webserver running on the same machine through StegoTorus (using SOCKS) and download a 4 MB file. The one-way delay is introduced in all 3 links. We also repeat each experiment varying the number of parallel StegoTorus circuits. We find the results to be promising (shown in Figure 5), i.e., the introduction of the JPEG steganography scheme introduces minimal additional overhead to StegoTorus. This is encouraging considering the fact that the JPEG steganography scheme is arguably superior to other proof-of-concept schemes currently implemented by StegoTorus.

Next, we compare the performance of the JPEG steganography scheme with each of the other steganography schemes implemented by StegoTorus (shown in Figure 5). Here, we vary the one-way link delay from 20-400 ms and fix the number of circuits to be 4. We find that the throughput of current JPEG steganography scheme falls in between that of PDF and JSON schemes. JavaScript performs best and SWF performs worst, while JSON and SWF schemes are least sensitive to link delay. We suspect that the relative insensitivity of JSON and SWF to link delays is because the file sizes transmitted by the StegoTorus server in these cases is much smaller than that of the other schemes. There is clearly room for additional optimization for the JPEG steganography

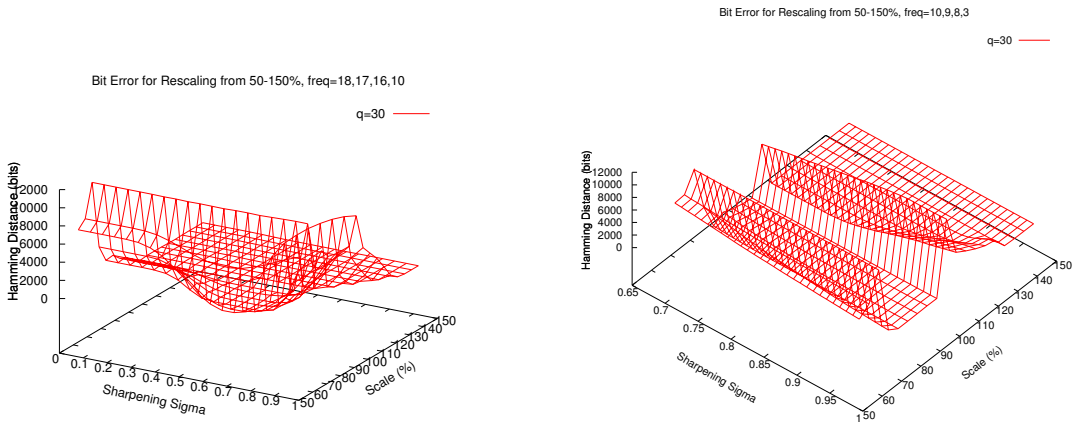

Figure 4: Left: Error as a function of sharpening sigma and image scale percentage. For this survey, frequency components 18,17,16, and 10 were used. Right: Error as a function of sharpening sigma and image scale percentage. For this survey, frequency components 10, 9, 8, and 3 were used. These are lower frequency components than in the previous plot, and exhibit a broader range of good performance.

scheme in terms of embedding data in more than four frequencies, tuning the quality levels etc. Evaluating these strategies in greater detail is future work.

7 Statistical Attacks and Limitations

We evaluate resilience of TRIST against three broad classes of attacks that have been employed against JPEG steganographic systems.

Histogram Divergence: (χ^2) Attack. The χ^2 attack uses first order statistics to detect the change in histogram between the normal and stegged image. Specifically, Westfeld and Pfitzmann developed an attack that detects LSB encoding variants using predictable pair-of-values (POVs) in the frequency histograms [38]. TRIST is not vulnerable to the POV χ^2 attack since it does not use LSB encoding. In addition, we performed some preliminary experiments to see whether there were any statistically significant differences in the distributions of frequency coefficients between steg and cover images, using default frequency selections. We performed these tests for each of the 64 frequency components and were not able to detect a difference with the Kolmogorov-Smirnoff test [17]. One possible explanation is that by default, TRIST restricts its operation to the most heavily quantized frequencies. These frequencies have very few categories to begin with, and the resulting post-steg distributions have a narrow peak centered about 0. Thus it may be difficult to use basic histogram-based statistical attacks to defeat TRIST.

Blockiness Detection. One attack that has proven successful against many steganography schemes is the self-calibrated blockiness measure proposed in [13]. Our approach may also be vulnerable to this attack, since the changes that we insert in the frequency domain are much more significant than just the LSB. We implemented the blockiness measure and message length estimator described in [13] and averaged the results over several cover images. We experimented with various quality levels for embedding, and found that if a message is embedded at a low quality (e.g., 30) and the resulting image is transcoded up to quality 90 (e.g., using 'convert'), the blockiness test no longer reliably determines message length. Figure 6 illustrates this effect for a range of message lengths from 0 to the maximum (around 39 KB).

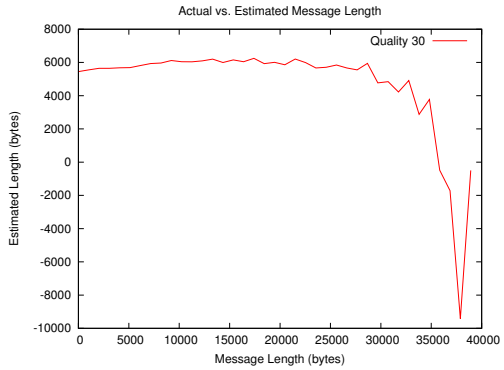

Figure 6: Message length estimates obtained using the blockiness measure, obtained by embedding the message at quality 30 and then transcoding up to quality 90 for a range of message lengths from 1-39 KB using 20 cover images from the BOSS dataset [2].

Blind Steganalysis. There has been a recent trend toward developing universal steganalysis tools that combine first and second order classifiers to detect steganographic images [23]. While we have not experimented against such systems, we anticipate that such attacks are likely possible against our system. However, these attacks rely on large feature vectors and tend to be computationally more expensive than prior attacks. Evaluating vulnerability to and building resilience to such attacks is future work.

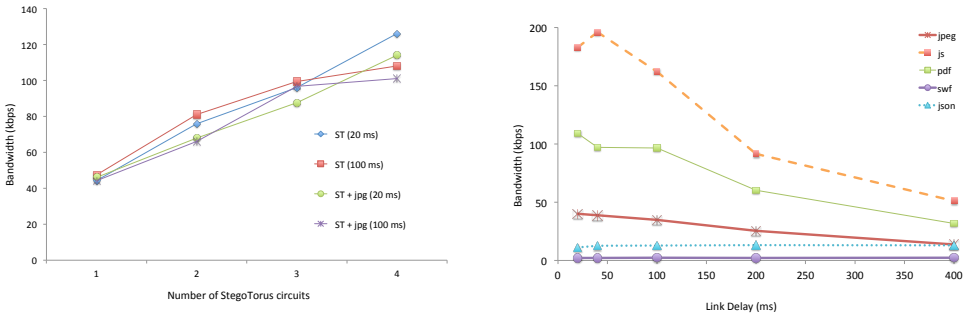

Figure 5: Left: Comparing StegoTorus thorughput with and without the JPEG steganographic scheme as we vary the number of circuits from 1 to 4 and the one-way propagation delay from 20 to 100 ms. JPEG steganography scheme has minimal impact on the performance of StegoTorus. Right: Comparing StegoTorus thorughput of various steganographic schemes (JavaScript, JSON, PDF, SWF and JPEG) as we vary the link delay from 20 to 400 ms.

8 Conclusion and Future Work

TRIST introduces a new twist to the standard steganography problem (i.e., transcoding resistance) and applies it to the censorship circumvention domain, which is an area of active research. An important challenge, associated with application of image steganography to this domain, is that of the channel bandwidth *(i.e., would the realized bandwidth be sufficient to sustain seamless web surfing?)*. We address this problem through the development of a new JPEG steganographic technique that provides improved robustness against automated transcoders by selectively modifying heavily quantized frequency components at low JPEG quality values. Our experimental evaluations demonstrate that we can robustly embed information across various images and this information survives a number of transformations, including transcoding to higher quality and rescaling of the image.

There are several potential areas of future work including (*i*) developing schemes that are resilient to other image transformations (e.g., rotations, smoothing etc.), (*ii*) integrating with other anti-censorship techniques such as Collage [3], MIAB [20] and FTE Proxy [7] and (*iii*) extending our strategies to JPEG-like encoding in other multimedia formats such as MPEG I-frames and shockwave flash files. Finally, steganography and censorship are both cat-and-mouse games and we anticipate that adversaries will develop new strategies to detect and disrupt our steganographic schemes. We view these as a natural evolution of the arms race and look forward to them as exciting opportunities to further improve our system.

9 Acknowledgments

We acknowledge helpful comments and feedback on this work from Drew Dean and Michael Walker. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) and Space and Naval Warfare Systems Center Pacific under Contract No. N66001-11-C-4022. 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 or Space and Naval Warfare Systems Center Pacific. Distribution Statement A: Approved for Public Release, Distribution Unlimited.

References

[1] M. Bardi, F. Bartolini, V. Cappellini, and A. Piva. A dct-domain system for robust image watermarking. *Signal Processing*, 66:357–372, 1998.

[2] P. Bas, T. Filler, and T. Pevný. Break our steganographic system — the ins and outs of organizing boss. In *Information Hiding, 13th International Workshop*, Lecture Notes in Computer Science, 2011.

[3] S. Burnett, N. Feamster, and S. Vempala. Chipping away at censorship firewalls with user-generated content. In *Proceedings of the 19th USENIX Conference on Security*, USENIX Security'10, 2010.

[4] C. Candan. A transcoding robust data hiding method for image communication applications. In *Proceedings of IEEE International Conference on Image Processing*, 2005.

[5] S. Chandra and C. S. Ellis. Jpeg compression metric as a quality-aware image transcoding. In *Proceedings of the 2Nd Conference on USENIX Symposium on Internet Technologies and Systems*, USITS'99, 1999.

[6] S. Chandra, A. Gehani, C. S. Ellis, and A. Vahdat. Transcoding characteristics of web images. In *SPIE Conference on Multimedia Computing and Networking*, 2001.

[7] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton. Protocol misidentification made easy with format-transforming encryption. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer Communications Security*, CCS '13, 2013.

[8] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. Karger. Infranet: Circumventing web censorship and surveillance. In *Proceedings of the 11th USENIX Security Symposium*, 2002.

[9] D. Fifield, N. Hardison, J. Ellithorpe, E. Stark, R. Dingledine, P. Porras, and D. Boneh. Evading Censorship with

Browser-Based Proxies. In *Privacy Enhancing Technologies*, 2012.

[10] D. Fifield, G. Nakibly, and D. Boneh. Oss: Using online scanning services for censorship circumvention. In *Privacy Enhancing Technologies*, 2013.

[11] T. Filler and J. Fridrich. Design of adaptive steganographic schemes for digital images. In *Proc. SPIE*, 2011.

[12] Flickr. http://www.flickr.com, 2014.

[13] J. Fridrich, M. Goljan, and D. Hogea. Attacking the outguess. In *ACM Workshop on Multimedia and Security*, 2002.

[14] J. Fridrich, M. Goljan, and D. Hogea. Steganalysis of jpeg images: Breaking the f5 algorithm. In *in 5th International Workshop on Information Hiding*, 2002.

[15] G. Gul and F. Kurugollu. A new methodology in steganalysis: Breaking highly undetectable steganography (hugo). In *Proceedings of 13th International Workshop on Information Hiding*, 2011.

[16] R. Hand, P. Bhagwat, R. LaMaire, T. Mummert, V. Perret, and J. Rubas. Dynamic adaptation in an image transcoding proxy for mobile web browsing. In *IEEE Personal Communications*, 1998.

[17] M. Hazewinkel. Kolmogorov-smirnov test. Encyclopedia of Mathematics, 2001.

[18] V. Holub and J. Fridrich. Digital image steganography using universal distortion. In *Proceedings of the First ACM Workshop on Information Hiding and Multimedia Security*, MMSec '13, 2013.

[19] A. Houmansadr, G. T. Nguyen, M. Caesar, and N. Borisov. Cirripede: Circumvention Infrastructure using Router Redirection with Plausible Deniability. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 187–200, 2011.

[20] L. Invernizzi, C. Kruegel, and G. Vigna. Message in a bottle: Sailing past censorship. In *Proceedings of the 29th Annual Computer Security Applications Conference*, ACSAC, 2013.

[21] Joint Photographic Experts Group. http://www.jpeg.org, 2014.

[22] J. Karlin, D. Ellard, A. Jackson, C. E. Jones, G. Lauer, D. P. Makins, and W. T. Strayer. Decoy Routing: Toward Unblockable Internet Communication. In *USENIX Workshop on Free and Open Communications on the Internet*, 2011.

[23] J. Kodovsky, T. Pevny, and J. Fridrich. Modern steganalysis can detect yass. In *SPIE, Electronic Imaging, Media Forensics and Security XII*, 2010.

[24] T. Lane and Independent JPEG Group. http://libjpeg.sourceforge.net, 2014.

[25] A. Latham. http://linux01.gwdg.de/ alatham/stego.html, 2014.

[26] P. Lincoln, I. Mason, P. Porras, V. Yegneswaran, Z. Weinberg, J. Massar, W. A. Simpson, P. Vixie, and D. Boneh. Bootstrapping communications into an anti-censorship system. In *2nd USENIX Workshop on Free and Open Communications on the Internet*, 2012.

[27] H. M. Moghaddam, B. Li, M. Derakhshani, and I. Goldberg. Skypemorph: protocol obfuscation for tor bridges. In *ACM Conference on Computer and Communications Security*, 2012.

[28] S. P. Mohanty and E. Kougianos. Real-time perceptual watermarking architectures for video broadcasting. *Journal of Systems and Software*, 84:724–738, 2011.

[29] A. Poljicak, L. Mandic, and D. Agic. Discrete fourier transform-based watermarking method with an optimal implementation radius. *Journal of Electronic Imaging*, 20(3), 2011.

[30] N. Provos. Defending against statistical steganalysis. In *10th USENIX Security Symposium*, pages 323–335, 2001.

[31] L. Rizzo. Dummynet: A simple approach to the evaluation of network protocols. *ACM Computer Communication Review*, 27:31–41, 1997.

[32] V. Santhi and D. A. Thangavelu. DWT-SVD combined full band robust watermarking technique for color images in YUV color space. *International Journal of Computer Theory and Engineering*, 1(4):424–429, 2009.

[33] A. Savant, N. Memon, and T. Suel. On the scalability of an image transcoding proxy server. In *International Conference on Image Processing*, 2003.

[34] K. Solanki, A. Sarkar, and B. S. Manjunath. Yass: yet another steganographic scheme that resists blind steganalysis. In *9th International Workshop on Information Hiding*, 2007.

[35] D. Upham. Jpeg-jsteg - modification of the independent JPEG group's JPEG software (release 4) for 1-bit steganography in jfif output files. http://www.tiac.net/usres/lorejwa/jsteg.htm, 1997.

[36] 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 *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, 2012.

[37] A. Westfeld. F5 – a steganographic algorithm: High capacity despite better steganalysis. In *4th International Workshop on Information Hiding*, pages 289–302. Springer-Verlag, 2001.

[38] A. Westfeld and A. Pfitzmann. Attacks on steganographic systems. In *Proceedings of the Third International Workshop on Information Hiding*, IH '99, pages 61–76, 2000.

[39] E. Wustrow, S. Wolchok, I. Goldberg, and J. A. Halderman. Telex: Anticensorship in the Network Infrastructure. In *Proceedings of the 20th USENIX Security Symposium*, pages 459–473, 2011.

[40] H. Zhang, H. Shu, G. Coatrieux, J. Zhu, Q. M. J. Wu, Y. Zhang, H. Zhu, and L. Luo. Affine legendre moment invariants for image watermarking robust to geometric distortions. *IEEE Transactions on Image Processing*, 20(8):2189–2199, 2011.